N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

"Made available under NASA sponsorship in the interest of early and wide dissemination of Earth Resources Survey Program information and without liability for any use made thereof."

E80-10307

JSC-14475

NASA CR 160757

"AS-BUILT" DESIGN SPECIFICATION FOR CLASSY CONVERSION

Job Order 71-593

TIRF (77-0072)

(E80-10307) AS-BUILT SPECIFICATION FOR CLASSY CONVERSION (Lockheed Electronics Co.) 261 p HC A12/MF A01 CSCL 09B

N80-30873

Unclas
G3/43 00307

Prepared By
Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200
For
EARTH OBSERVATIONS DIVISION
SPACE AND LIFE SCIENCES DIRECTORATE

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

September 1978

LEC-12743

JSC-14475

"AS-BUILT" DESIGN SPECIFICATION
FOR
CLASSY CONVERSION

Job Order 71-593

TIRF (77-0072)

Prepared by

P. J. Aucoin
C. Horton

APPROVED BY

John A. Rainey

John A. Rainey, Supervisor
Scientific Applications Section

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division
Science and Applications Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

September 1978 LEC-12743

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

1. SCOPE

This document describes the conversion of the CLASSY clustering/ classification program from the UNIVAC EXEC 2 in JSC Bldg 12 to the IBM 370/148 computer at LARS, West Lafayette, Indiana.

The converted program is written exclusively in Fortran IV-G. The conversion approach has been similar to that followed in the EOD-LARSYS system conversion.

Several program enhancements and/or changes have been incorporated as a result of information made available during the conversion process.

2. APPLICABLE DOCUMENTS

- TIRF 77-0072
- "As-Built" Design Specification, "CLASSY Program Modification," TIRF 77-0055, JSC-13986, LEC-12185, April 1978.
- "Program Documentation for Modification to the CLASSY Program," JSC-12602, LEC-10481, April 1977.
- Technical Memorandum, "Final Acceptance Test Plan for the EOD-LARSYS conversion," Ref: 646C-13, March 1978.

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

The CLASY clustering program, as modified, is operational on the Univac 1110 (or 1108) under the EXEC8 operating system. The program utilizes the Univac Fortran V compiler (the original program utilizes the Univac reentrant Fortran compiler, RFOR), the Univac assembler (for assembly routines FREE, GET and LOCK), and the Univac system random file access routines RINIT, RREAD, and RWRITE.

3.2 SOFTWARE DESCRIPTION

The CLASY system of subprograms was originated by Dr. Michael Rassbach, a post-doctoral contractor for NASA-JSC, Earth Observations Division (TF).

The purpose for his development of CLASY was the implementation of an iterative statistical clustering algorithm which had theoretical promise for application to classification of earth resources (image) data acquired from the LANDSAT (formerly ERTS) satellite.

These modifications to CLASY are (1) to improve the program execution time, especially input/output overhead, in order to make it feasible to test and evaluate the CLASY program, (2) to implement a randomized input data scrambling technique which was obtained from Rice University by the Earth Observations Division's Research, Test and Evaluation (RT&E) Branch (TF3). The scrambling of input data vectors is a necessity in CLASY due to the algorithms; sensitivity to correlated data values and (3) to allow users to specify more than 4 channels for classification.

The CLASY system of subprograms consists of the main driver program, CLASY, and 56 subprograms, not including the Univac system routines utilized by the program. Four of the subrprograms (LOCK, GET, FREE, and BYTRAN) were originally programmed Univac assembly language (SLEUTH II), the remaining subprograms and the main program, CLASY, are converted to the Univac Fortran V language from the former (original) Univac reentrant Fortran language, RFOR.

The driver program for the clustering system is CLASY. The data handling subprograms for the system are READTP and STATIS which were reprogrammed to implement the required data input improvement and to implement the randomized data scrambling technique. READTP provides the data setup for acquisition on demand by the iterative statistical subprogram, STATIS. STATIS initiates the clustering procedure, operating on one pixel (data vector) at a time in setting up clusters and making the cluster split/combine decisions. Each pixel is examined 10 times by STATIS during the clustering procedure.

To implement the required modifications and the modifications suggested by the program originator to improve the program's reliability, the following routines in the CLASY clustering system received changes: CLASY, CLASY1 (deleted), ADJUST, LOCK (formerly LOC), CLPRM, (formerly CLPR) CLDUMP, MISH, SETUP9, CMBK10, ELIM, MULTI, SEPER, TR, STATIS, CLUST.

The modified subprograms are discussed in the order of their use by the CLASY system.

The overall CLASY system is flowcharted in Appendix A. Listings of the modified routines are shown in Appendix B. Sample output from the CLASY system is shown in Appendix C.

3.2.1 SOFTWARE COMPONENT NO. 1 (CLASY)

3.2.1.1 Linkages

CLASY is the driver program of the CLASY clustering system. CLASY calls SETUP9, READTP, MULTI, and CLUSMP.

3.2.1.2 Interfaces

The common blocks INFORM, CLUSTR, CLUS, MISC, and STPAR and calling arguments are used in the program CLASY as interfaces with other routines in the clustering system.

3.2.1.3 Inputs

CLASY calls SETUP9, which reads the input supervisor (control) cards. The supervisor cards and their functions are described in the discussion of SETUP9 (section 3.2.2.6).

The required input to the CLASY program consists of one tape (or file) containing the multichannel image data and the special-format card input.

The image data tape (file) is presumed to be in either of two specific formats--either "LARSYS II" format or "UNIVERSAL" format. The tape (file) reading program in CLASY, TAPERD, accepts either of these formats and self-determines the correct method of reading the data.

3.2.1.4 Outputs

The output by CLASY is all line-printer output. Interim printout of statistical parameters and diagnostic data is provided during the iterative cluster-forming process.

The final output is a "map", with a symbolic representation of area clustered, with each pixel of the area classified using the statistics (mean and covariance) from final cluster set determined by CLASY. The symbols on the "map" represent the cluster (=class) which is the most likely parent distribution for the given pixel. The "map" is output by subprogram CLUSMP.

Sample output is shown in Appendix C.

3.2.1.5 Storage Requirement

Storage used: Code = 36_8 Data = 42112_8

3.2.1.6 Description

CLASY is the driver program for the clustering routines. It was rewritten to (1) enable CLASY1 to retrieve large blocks of data (dimensioned ARRAY (20 000)) from drum, (2) to pre-calculate the amount of data RREAD (drum read

subroutine) will read from drum, (3) to make DATAB (the array containing the scrambled data) and PV (the array passed to the clustering routine, STATIS) to reside in the same locations as the large data array, "ARRAY" and (4) to selectively skip calculations on clusters which have subclusters. All other logic in the program remains the same as in the pre-modified version of CLASY.

3.2.1.7 Flowchart

See Appendix A.

3.2.1.8 Listings

See Appendix B for program.

3.2.1.9 Restrictions

The known restrictions inherent in the program are (1) the program will not successfully execute with only one channel, (2) a data vector containing a zero value in the channel of interest will cause an error termination of the program's execution, (3) the size of the original image data set read from the input tape (or file) and placed on drum must be containable in 1,310,717 locations of drum storage available to the random access routines (RINIT, RREAD, and RWRITE).

3.2.2 SOFTWARE COMPONENT NO 2 (SETUP9)

3.2.2.1 Linkage

SETUP9 is called from CLASY. SETUP9 calls NXTCHR and NUMBER, which are entry points in subroutine FIND.

3.2.2.2 Interface

Interface is accomplished through calling arguments and the following common blocks: INFORM, SUPCUM, and CLUSTR.

3.2.2.3 Inputs

See Appendix C.

3.2.2.7 Flowchart

See Appendix A.

3.2.2.8 Listings

See Appendix B for the modified program listing.

3.2.3 SOFTWARE COMPONENT NO. 3 (READTP)

3.2.3.1 Linkage

READTP is called from CLASY. READTP calls RREAD, RWRITE, CMERR, and ZOR (Function ZOR is the random number generator used in the data scrambling technique), RINIT, TAPHDR, LAREAD, FLDINT, LINERD, FDLINT and ERTRAN.

3.2.3.2 Interface

Interface is accomplished through calling arguments and the following common blocks: INFORM, CLUSTR, CLUS, MISC, and STPAR.

3.2.3.3 Inputs

Image data tape described in 3.2.1.3

3.2.3.4 Output

READTP outputs the following error messages:

3.2.3.5 Storage Requirement

Storage used: Code = 1652_8 Data = 427_8

3.2.3.6 Description

READTP performs the input image data-handling function for the CLASY clustering system and makes the image data available on disk to the iterative statistical subprogram, STATIS. The original image data from the area on

the input tape (file) which has been designated by the input field-definition card(s) is prestored on drum as one continuous block of data. The Univac random file access routines -- RINIT, RWRITE -- are utilized to place the input data on the drum as the data is read from the input tape (file) by the TAPERD subprogram.

READTP precomputes the base addresses for three data-buffering arrays in core-storage. One half of data-buffering array, ARRAY is used to retrieve a block of original image data from the prestored drum.

The other half of array, ARRAY is used to contain a set of integers -- 1,2,3,...,N where N=the number of data vectors in the original image array. Upon completion the first half of the array will contain a set of data vectors obtained from the original image set, but stored such that each data vector's original spatial location is randomly rearranged. The data scrambling technique utilized in rearranging the data vectors was obtained from Rice University via RT&E and is the required modification implemented by this modification to the CLASY clustering system. The implementation of the randomized data scrambling technique is performed as follows:

(1) Given an array of original image data vectors, and an array, of integers -- 1,2,3....,N with N = the number of image data (2) Scramble the

(2) Scramble the elements of A

a. Obtain a random number, Z_i, from the uniform random number generator;

$$Z_i = ZOR(0), \quad 0. \leq Z_i \leq 1.0$$

b. Multiply the random number Z_i, by N, the largest integer in A;

$$IX_i = N \times Z_i + 1$$

c. Using IX_i as an index, scramble the integers in A as follows:

3.2.2.4 Output

SETUP9 prints out a summary of the input to CLASY and also prints an error message on the line printer, if an invalid input card is detected. If an error is detected, SETUP9 prints the following message "INVALID INPUT CARD--IGNORED", processing continues.

3.2.2.5 Storage Requirement

Storage used: Code = 275_8 Data = 157_8

3.2.2.6 Description

SETUP9 reads and analyzes all cards input to the CLASY program. SETUP 9 was modified to add "0" (zero) as a new symbol to the symbol array "SMBLS". The following control cards are input to the modified CLASY program, to be analysed by SETUP9. In all cards, the "keyword" begins in card column 1, and any parameters on the card are placed from card columns 11 through 72, inclusive.

1. "CHANNEL" CARD (i.e., "CHANNEL 1,5,9,13")

The "CHANNEL" card specifies the channel numbers to be used in clustering the multi-channel data vectors. At present the maximum number of channels allowed to specify is eight. The identification "CHANNEL" starts in column 1, and the actual channel numbers, separated by commas, start in card column 11, and must be terminated by column 72.

2. "PRINT" CARD (i.e., "PRINT 1,3,3")

The "PRINT" card specifies how to print the cluster map. The identification "PRINT" starts in column 1, the actual print parameters start in card column 11, separated by commas, and ending by column 72.

3. "HED1" card

4. "HED2" card

These two cards may be used to specify any arbitrary heading for the printer output, including the cluster map. Any alphanumeric characters put into card columns 11-72 of these two cards will be output as a page heading.

5. "NPOS" card

This card, used previously in the unmodified CLASY program, specified the number of positions to skip to read a pixel point (i.e. to "scramble" the data). This card is not used in the modified CLASY program.

6. "NPTS" card

This card, used previously in the unmodified CLASY program, specified the number of pixels to retrieve from the data set at each point. This card is not used in the modified CLASY program. (NPTS was used in the original programs data scrambling technique).

7. "DATE" card

This card is used to specify the date or any eight characters. Will be printed at the upper right hand corner of each page of printer output.

8. "COMMENT" card

The "COMMENT" card is equivalent in use and format with the "HED1" and "HED2" cards, described above.

9. "*END*" card

This card specifies the end of all supervisor (control) card (described above) input to CLASY. This card is a mandatory input to CLASY, to initiate the clustering process.

$$TEMP = A_i$$

$$A_i = IX_i$$

$$A_{IX_i} = TEMP$$

d. Execute the above procedure (a-c) N times, with

$$i = N, N-1, N-2, \ldots, 1$$

e. Create the new (scrambled) set of data vectors, A_s, as follows;

$$\text{for } i = 1, 2, 3, \ldots, N$$

$$A_{s_i} = A(j), \text{ where } j = A_i$$

The scrambled data set, is made available in large blocks for retrieval on demand from the statistical clustering subprogram, STATIS. The scrambling of data for STATIS is necessitated by the clustering algorithm's sensitivity to correlated data.

The data buffering technique as described accomplished the first objective of the modifications to the CLASY clustering system--namely, the improvement of program execution time.

3.2.3.7 Flowchart

See Appendix A.

3.2.3.8 Listings

See Appendix B for program listing.

3.2.4 SOFTWARE COMPONENT NO. 4 (MULTI)

3.2.4.1 Linkage

MULTI is called from CLASY. MULTI calls DATFIX, ALFREE, CLINIT, STATIS and CLDUMP.

3.2.4.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, INFORM and CLUSTR.

3.2.4.3 Inputs

None.

3.2.4.4 Output

None.

3.2.4.5 Storage Requirements

Storage used: Code = 105_8 Data = 11_8

3.2.4.6 **Description**

MULTI calls the routines to initialize the clustering algorithm. MULTI was modified to make use of unused areas in core. ARRAY (EXARAY) was previously dimensioned but never used. A method was devised to make use of this unused core such that the PV array and DATAB array (array utilized in STATIS and the array containing scrambled data vectors) be made to utilize the same area of core (i.e., the PV and DATAB arrays were made equivalent storage areas, in ARRAY (EXARAY)).

3.2.4.7 Flowchart

See Appendix A.

3.2.4.8 Listings

See Appendix B for program listing.

3.2.5 SOFTWARE COMPONENT NO. 5 (STATIS)

3.2.5.1 Linkages

STATIS is called by MULTI. STATIS calls DISC, CLASY2, CORECT, DOTSQ, VPV, VMTV, MPVS, ADJUST, CLDUMP, and EXP.

3.2.5.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, CLUSTR, and RAND.

3.2.5.3 Inputs

None.

3.2.5.4 Outputs

STATIS outputs two warning messages. They are: "***WARNING ON THE__ INDEX(KL)=", "**SUSPECTED BAD DATA POINT --STATIS**IDO=__,ROOT__, VECTOR__"

3.2.5.5 Storage Requirements

Storage used: Code = 1305_8 Data = 306_8

3.2.5.6 Description

STATIS takes each input data vector and classifies it on a fractional, probabilistic basis. It then updates the various statistical parameters associated with the classes (clusters) indicated and checks to see if any of these classes is potentially two. Those which are will be referred to the routine "SPLIT". The one modification made to STATIS was the re-dimensioning of the PV array, which contains the data vectors to be clustered.

3.2.5.7 Flowchart

See Appendix A.

3.2.5.8 Listings

See Appendix B for program listing.

3.2.6 SOFTWARE COMPONENT NO. 6 (ADJUST)

3.2.6.1 Linkage

Adjust is called from STATIS. ADJUST calls GET, TR, DOTSQ, SQMTX, MINV, UNIF, CLPR, TRIMTX, DENCAL, SPLIT, FREE, CLDUMP, SEPER, SUBLIM, ELIM, CORECT, JOIN, APRIOR, SQRT, ALDG, EXP, and XPRI.

3.2.6.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, CLUSTR, and JOINPR.

3.2.6.3 Inputs

None.

3.2.6.4 Outputs

ADJUST prints out three brief messages concerning statistical information and three error messages. They are; "ADJUST__ WEIGHT__ WAS__ SPFAC__ CHANGE__ __ __", "STATISTICS: TRACE__ SKEW__ KURT__ TESTS (SPLIT >0): __ __ __", "###HAVE SPLIT__ WEIGHT__ SUBS__ __", "W/OVOL ERROR IN ADJUST: KL,W,NEW W,VOL __ __ __ __", "***EXTRAPOLATION PROBLEM IN ADJUST: ITER, INDEX(KL), VOLIN, OVOL, CVOL__ __ __ __ __", "LOG ERROR IN ADJUST: I, IM, KL, K/VRIN= __ __ __ __ __ __"

3.2.6.5 Storage Requirements

Storage used: Code = 2406_8 Data = 354_8

3.2.6.6 Description

One modification made to ADJUST was to eliminate the use of subscripted subscripts for those arrays which previously presumed the RFOR (reentrant Fortran) compiler. The change was made to enable the routine to compile under the Univac 1100-EXEC 8 Fortran V compiler. The second change made was that input to the "ALOG" routine is forced to be positive by use of the absolute value of the input being sent to "ALOG". The reason for the change was than an occasional negative value was being sent to "ALOG", causing an error termination. The routine previously referenced a dimensioned variable in several calling statements causing an error during program execution. These errors were corrected. Other modifications which were suggested by the CLASY program originator, Dr. Mike Rassbach, are changes in the calculation of EXF, WADJ, DCORR and ALINK.

3.2.6.7 Flowchart

See Appendix A.

3.2.6.8 Listings

See Appendix B for program listings.

3.2.7 SOFTWARE COMPONENT NO. 7 (TR)

3.2.7.1 Linkage

TR is called from ADJUST.

3.2.7.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.7.3 Inputs

None.

3.2.7.4 Outputs

None.

3.2.7.5 Storage Requirements

Storage used: Code = 124_8 Data = 35_8

3.2.7.6 Description

The modification made to TR was the elimination of subscripted subscripts, to enable this routine to be compiled under the Fortran V compiler on the Univac 1110 (EXEC 8 system). The subscripted subscript notation was a feature of the reentrant Fortran compiler RFOR, utilized by the program originator in development of CLASY.

3.2.7.7 Flowchart

See Appendix A.

3.2.7.8 Listings

See Appendix B for program listing.

3.2.8 SOFTWARE COMPONENT NO. 8 (CLPRM)

3.2.8.1 Linkage

CLPRM is called from CLDUMP, ADJUST, SEPER and JOIN. CLPRM calls GET, LOCK, SQMTX, MINV, and FREE.

3.2.8.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.8.3 Inputs

None.

3.2.8.4 Outputs

CLPR prints out seven messages concerning statistical information. These messages are: "CLUSTER__ INDEX__ PROPORTION__ W*__ SPLIT__ WEIGHT__ WAS__ ADJUST__ TO__ PROPORTION: PROP__ CIN__ CTUT__ OLD PROP__ CIN__ ODEN__ DIFFER__ VOLUME__ ROUT__ DCON__", "LOCATION__ LINK__ SUBS__ SUPER__ __ SYMBOL__", "NET PROB__ DIRECT__ CUMS__ __", "CUMS__ __", "MEAN__ __ __ __", "KURT(*W)__ __ __ __", "OLD COVARIANCE__ __ __ __ __"

3.2.8.5 Storage Requirements

Storage used: Code $=736_8$, Data $=364_8$

3.2.8.6 Description

The modifications made to CLPR to form CLPRM were (1) the elimination of subscripted subscripts (2) a reference to one of CLASY's clustering routines, named "LOC", caused ambiguity, because there existed a Univac system routine with the same name. Therefore this reference and all other references to "LOC" were changed to "LOCK". The originator of CLASY, Dr. Michael Rassbach, provided changes to be made in CLPR. One modification suggested by Dr. Rassbach was the changing of a format statement so that the printer will skip a line before writing the statistical mean.

3.2.8.7 Flowchart

See Appendix A.

3.2.8.8 Listings

See Appendix B for program listing

3.2.9 SOFTWARE COMPONENT NO. 9 (CLDUMP)

3.2.9.1 Linkage

CLDUMP is called from CLASY1, MULTI, CLASY2 and ADJUST. CLDUMP calls ISPLIT and CLPR.

3.2.9.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, and CLUSTR.

3.2.9.3 Inputs

None.

3.2.9.4 Outputs

A header concerning the dump of the clusters and a debug statement giving the value of the number of the split cluster and the value of the print parameter.

3.2.9.5 Storage Requirements

Storage used: Code = 153_8, Data = 33_8

3.2.9.6 Description

The modifications made to CLDUMP were the elimination of the logical "IF" statement that checked the value of ISPLIT and the print variable, PROUT, before the call is made to CLPR. A debug printout statement was added to allow printout of these variables for checkout purposes.

3.2.9.7 Flowchart

See Appendix A.

3.2.9.8 Listings

See Appendix B for program listing.

3.2.10 SOFTWARE COMPONENT NO. 10 (ELIM)

3.2.10.1 Linkage

ELIM is called from ADJUST. ELIM calls SUBLIM and TRFREE.

3.2.10.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.10.3 Inputs

None.

3.2.10.4 Outputs

ELIM prints out two messages, one statement when a cluster has been eliminated and the other is an error message when a structural error has occurred. These messages are: "###ELIMINATE__ LINK, LSUBS, LSUPER = __ __ __", "**STRUCTURAL ERROR AT ELIM: KEL, KFAITH, KOLD, INIT__ __ __ __ __".

3.2.10.5 Storage Requirements

Storage used: Code = 176_8, Data = 46_8

3.2.10.6 Description

The modifications made to ELIM were the eliminations of subscripted subscripts in arrays to allow the routine to be compiled under the Fortran V compiler of the Univac 1110-EXEC 8 system.

3.2.10.7 Flowchart

See Appendix A.

3.2.10.8 Listings

See Appendix B for program listings.

3.2.11 SOFTWARE COMPONENT NO. 11 (SEPER)

3.2.11.1 Linkages

SEPER is called from ADJUST. SEPER calls CLPR, DGNCAL, and FREE.

3.2.11.2 Interface

Interface is accomplished through the following common blocks: CLUS, MISC, and STPAR.

3.2.11.3 Inputs

None.

3.2.11.4 Outputs

SEPER prints one statement, which is printed whenever a cluster is split. This statement is : "###SEPARATE__ SUPER, SUBS__ __ SPFAC__"

3.2.11.5 Storage Requirements

Storage used: Code = 237_8 Data = 45_8

3.2.11.6 Descriptions

The modifications made to SEPER were the elimination of subscripted subscripts in arrays, to enable the routine to be compiled under the Fortran V compiler of the Univac 1110-EXEC 8 system.

3.2.11.7 Flowchart

See Appendix A.

3.2.11.8 Listings

See Appendix B on program listing.

3.2.12 SOFTWARE COMPONENT NO. 12 (LOCK, formerly "LOC)

3.2.12.1 Linkages

LOCK is a function subprogram and is called from CLPR, and CLPRM.

3.2.12.2 Interface

Interface is accomplished through the function arguments.

3.2.12.3 Inputs

None.

3.2.12.4 Output

None.

3.2.12.5 Storage Requirements

Storage used: Code ≈ 16_{10}

3.2.12.6 Descriptions

The modification made to LOCK (formerly "LOC") was the changing of the external reference name from 'LOC' to 'LOCK'. The name 'LOC' is also the name of a Fortran V system routine which caused ambiguity between the two routines during execution of CLASY. Changes were made to all other routines that referenced this routine, to change the reference to be "LOCK" rather than "LOC".

3.2.12.7 Flowchart

See Appendix A.

3.2.12.8 Listings

See Appendix B for program listing.

3.2.13 SOFTWARE COMPONENT NO. 13 (MISH)

3.2.13.1 Linkages

The Fortran V procedure, MISH, is included in the following routines: ACOM, ADJUST, ALFREE, AMSQ, APRIOR, CBLO, CLASY, CLDUMP, CLINIT, CLPRM, CLUSMP, CLUST, CORECT, DATFIX, DOTSQ, EIGROT, ELIM, ISPLIT, JOIN, MINV, MLT, MPVS, MTVEC, MULTI, MVEC, SEPER, SPLIT, SQMTX, STATIS, STOFLO, SUBLIM, TR, TRFREE, TRIMTX, VMTV, and VPV.

3.2.13.2 Interface

The Fortran V procedure, MISH, is used as the interface for the following common blocks: CLUS, MISC, and STPAR.

3.2.13.3 Inputs

None.

3.2.13.4 Output

None.

3.2.13.5 Storage Requirements

Storage used: None

3.2.13.6 Descriptions

The modification made to the proc, MISH, was the changing of the symbol array 'NSYMB' dimension from 11 to "12" (i.e., NSYMB(12)). This was done because another symbol was added to the large symbol array, 'SYM'.

3.2.13.7 Flowchart

See Appendix A.

3.2.13.8 Listings

See Appendix B for program listing.

3.2.14 SOFTWARE COMPONENT NO. 14 (CMBK10)

3.2.14.1 Linkages

The Fortran V procedure, CMBK10, is included in the following routines: ADJUST, CLASY, CLDUMP, CLINIT, CLUSMP, JOIN, MULTI, READTP, SETUP9, and STATIS.

3.2.14.2 Interface

The Fortran V procedure, CMBK10, is used as the interface for the common block , CLUSTR.

3.2.14.3 Inputs

None.

3.2.14.4 Output

None.

3.2.14.5 Storage Requirement

Storage used: None

3.2.14.6 Description

The modifications made to the Fortran V procedure CMBK10 were as follows: (1) The changing of the parameter, MAXPOP, value from 60 to 61. This was done to increase the dimension of the symbol array to account for the addition of another symbol to the array. The symbol added to the array was the character '0' (zero). (2) The maximum number of channels was changed from 4 to 16.

3.2.14.7 Flowchart

See Appendix A.

3.2.14.8 Listings

See Appendix B for program listing.

3.2.15 SOFTWARE COMPONENT NO. 15 (CLUST)

3.2.15.1 Linkage

CLUST is called from CLUSMP. CLUST calls ISPLIT, CORECT, DOTSQ, and EXP.

3.2.15.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, BIGCOM.

3.2.15.3 Inputs

None.

3.2.15.4 Outputs

None.

3.2.15.5 Storage Requirements

Storage used: Code = 332_8, Data = 112_8

3.2.15.6 Description

3.2.15.7 Flowchart

See Appendix A.

3.2.15.8 Listing

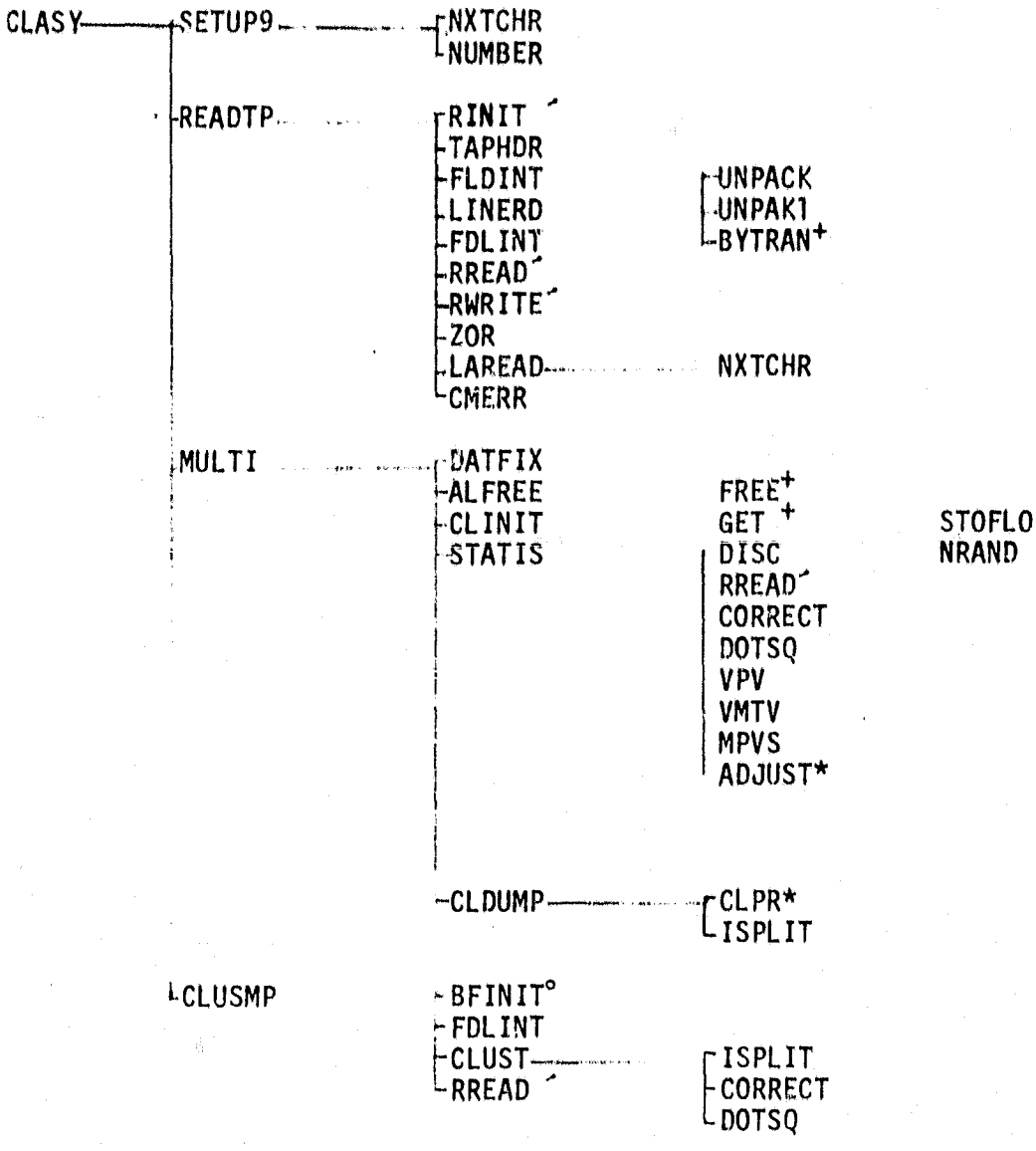
See Appendix B for program listing.

4.0 OPERATION

The source code and relocatable elements for the CLASY system are on tape 12667, located in Building 12, Johnson Space Center, Houston, Texas. The CLASY clustering program is operational on the Univac 1110 (or 1108) under the EXEC8 operating system.

APPENDIX A

CLASY SYSTEM FLOWCHART

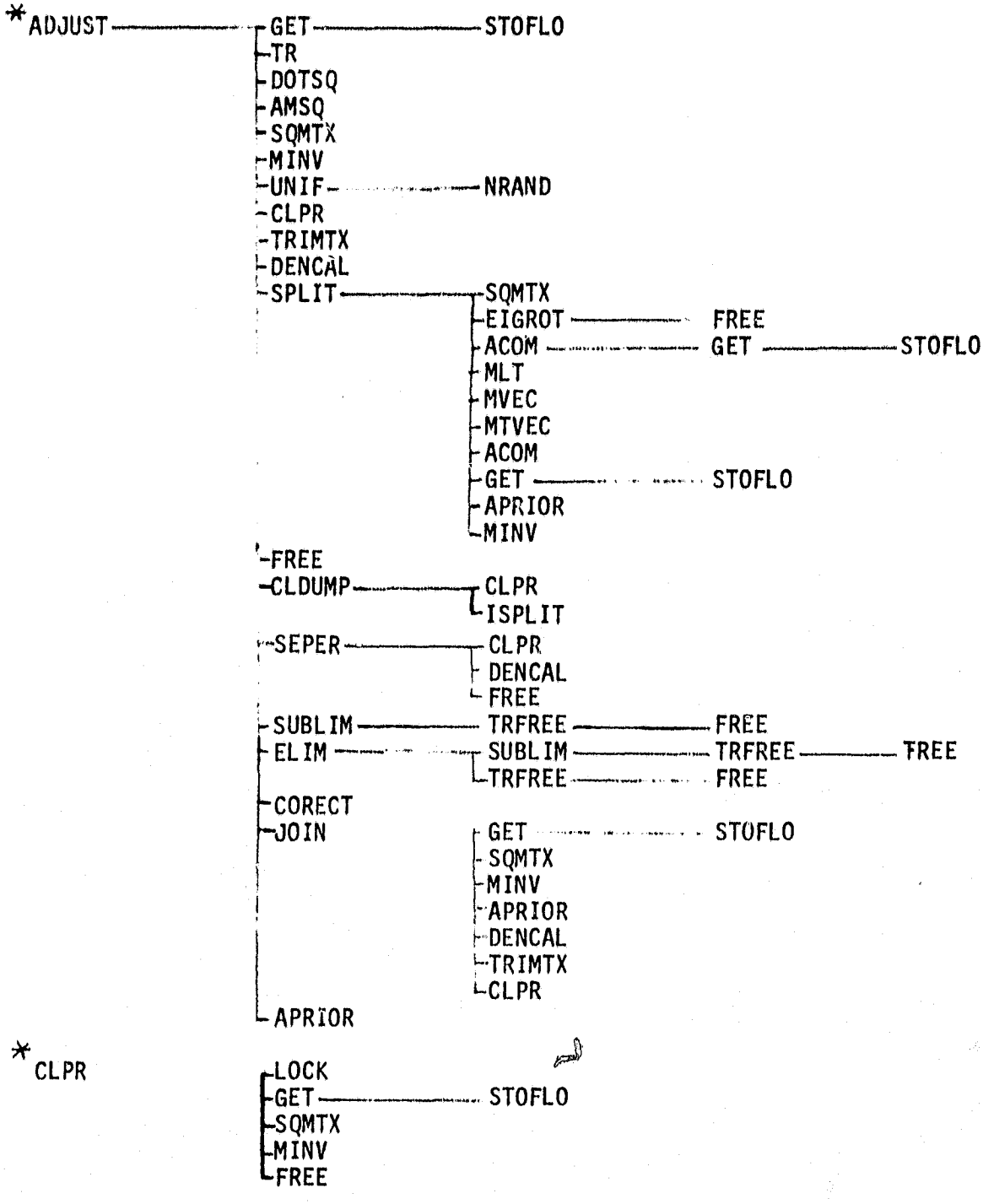


\+ = Univac Assembly Language Routine

0 = Fortran internal subroutine

´ = Univac Random I/O (random file access) routine

* = Remainder of flowchart on page A-3



ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

CLASY SYSTEM SUBROUTINES

```
@PDP,FL CMBK10,CMBK10,CMBK10
PDPT 10 RL70-6 04/18-01:27:29-10.0)
PE 00 01        CMBK10 PROC
   00 02              PARAMETER MAXPOP=61
   00 03        C
   00 04              COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(MAXPOP) ,
   00 05             1               LNCAT, PRNT(4) , KLBC , PRTME, PROUT, TOTPIX,
   00 06             2  DUMP,SCRAM1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD
   00 07             3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1
   00 08        C
   00 09              INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT                -01
   00 10             1 ,CLSNAM
   00 11        C
   00 12              EQUIVALENCE ( NIT,PRNT(4) )
   00 13        C
   00 14              DATA PRNT(4)/10/
   00 15        C
   00 16        C  BUFPIX = SIZE OF 'ARRAY'/NO. OF CHANNELS                                    CM
   00 17        C  CLSNAM =
   00 18        C
   00 19        C  IBEGIN = FIRST AVAILABLE WORD ON FAST STORAGE                               CM
   00 20        C  IPT = 1ST AVAIL WORD IN SCRATCH AREA 'ARRAY'
   00 21        C  KLBC =
   00 22        C
   00 23        C  LBUFD = LEGTH OF LAST RECORD IN ORIG DATA AS COPIED TO WORD AD DRUM         CM
   00 24        C  LNCAT =
   00 25        C
   00 26        C  NBUFSD = SIZE OF ORIG DATA/SIZE OF BUFFER 'ARRAY'                           CM
   00 27        C  NDUMP = SIZE OF AVAILABLE AREA IN 'ARRAY'
   00 28        C  NIT = NO. ITERATIONS REQUESTED FOR SCRAMBLED DATA SET, DEFLT TO 10          CM
   00 29        C  NOFLD =
   00 30        C
   00 31        C  PROUT = PRINT FLAG
   00 32        C
   00 33        C  PRTME = PRINT FLAG
   00 34        C
   00 35        C  SCRAM1 = 1ST WORD OF AVAIL FAST STORAGE + LENGTH OF ORIG DATA UNLESS        CM
   00 36        C       ORIG DATA MOVED THEN 1ST WORD OF AVAIL FAST STORAGE                    CM
   00 37        C  SUFTOT = NO. WORDS AVAIL IN SCRATCH AREA 'ARRAY'                            CM
   00 38        C  SYM = WORD ARRAY OF SYMBOLS
   00 39        C  TOTPIX = TOTWRD/MQ,  INITIALIZED IN STATUS
   00 40        C  TOTWRD = NO. WORDS IN ORIGINAL DATA ON DRUM                                 CM
   00 41        C
   00 42         END CMBK10

END PDP  ERRORS : NONE
```


ORIGINAL PAGE IS OF POOR QUALITY

```
@PCP,FL MISH,MISH,MISH
PDPT10 RL70-6 04/18-01:27:31-10,0)
PE00 01        MISH   PROC
   00 02        C
   00 03              PARAMETER NTBSIZ=32
   00 04        C    NTBSIZ MUST BE A POWER OF TWO.  (NTB & NTBSIZ ARE USED IN THE
   00 05        C          STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREE.)
   00 06        C
   00 07        C       THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF
   00 08        C         SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.
   00 09        C
   00 10              PARAMETER NARRS=3
   00 11              PARAMETER NVECS=3
   00 12              PARAMETER NSCALS=25
   00 13              INTEGER JUNK(12), NTB(NTBSIZ), LINK(999)
   00 14        C
   00 15        C   ** TYPE STATEMENTS CANNOT BE USED FOR DIMENSION STATEMENTS ON IBM 370
   00 16              REAL W(1),PORAT(1),VOLIN(1),VOLPT(1),DCON(1),PPASS(1),PCOND(1)
   00 17              INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)
   00 18              REAL VRIN(475),GEN(999),GREF(999),ALINK(1)
   00 19              REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(1),OW(1),SPFAC(1)
   00 20              REAL PRIRCM(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)
   00 21        C   THE FOLLOWING STRUCTURES THE CLUSTER NODES.  'GEN' IS
   00 22        C      USED LATER AS THE BASE FOR ALL THE ARRAYS.  (SEE CLINIT)
   00 23        C           'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,
   00 24        C                RATHER THAN IN CALLING SEQUENCES.
   00 25              EQUIVALENCE (LINK(31),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),
   00 26             1    INDEX(27),PCUM(26),PRIRCM(25),CIN(24),CTOT(23),
   00 27             2     PROP(22),SPFAC(21),WADJ(20),W(19),OPROP(18),OW(17),
   00 28             3     VOLIN(16),VOLRT(15),DCON(14),PORAT(13),DISS(12),PPASS(12),
   00 29             4     PST(11),OCIN(10),PCOND(7),VRIN(7),GEN(7),OPRIOR(9),ODEN(8),
   00 30             5     GREF(8),NTB(31))
   00 31              EQUIVALENCE (LINK(1),ALINK(1))
   00 32        C
   00 33              COMMON/CLUS/  JUNK,NARL,NTOP,NTBSZM,NWANT,LINK
   00 34        C
   00 35        C
   00 36        C   WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE
   00 37        C      ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV
   00 38        C      FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED
   00 39        C      FOR CONVENIENCE. (FOR EXAMPLE, GEN(LSUM+KL+I) IS THE
   00 40        C      VECTOR SUM(KL+I) .)
   00 41              INTEGER MXAR(31),LR(NARRS),LV(NVECS)
   00 42        C   ** TYPE STATEMENTS CANNOT BE USED FOR DIMENSION STATEMENTS ON IBM 370
   00 43              EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),
   00 44             1   (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)
   00 45        C
   00 46              COMMON /MISC/ MO,MM,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT,
   00 47             1   AMO,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH,
   00 48             2   INDXVL,WFAC,NPTSO,PORATH,SPMVTH,DWFAC,GRACTM,AMOFAC,
   00 49             3    AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,
   00 50             4   BETTER,MODE,CORLEN,SPCOR
   00 51        C
   00 52              REAL PACCEL(2),MACCEL(2),VACCEL(2)
   00 53        C
   00 54              COMMON /STPAR/ WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,
   00 55             1    PACCEL,MACCEL,VACCEL
```

```
00056       C     MQMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)
00057       C
00058       C
00059             PARAMETER MQMAX = 8
00060       C                                                              NEW
00061       C       MQTMAX=(MQMAX*(MQMAX+1))/2                             -01
00062             INTEGER GET
00063       C
00064             PARAMETER MQTMAX = 36
00065       C                                                              NEW
00066        END MISH                                                      -01
END FDP  ERRORS : NONE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S ACOM,ACOM,ACOM
FOR SOE3-C4/18/78-01:27:32 (0,0)

   SUBROUTINE ACOM      ENTRY POINT 000153

   STORAGE USED: CODE(1) 000172; DATA(0) 000046; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003   CLUS   002017
    C004   MISC   000113
    C005   STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000031 1336      0001    000192 136G      0001    000194 142G      0003 R 000020 ALINK     0004    000064 AMM
    C004    000075 AMOFAC    0004    000077 AMOMAX    0004    000076 AMOMIN    0004   000100 AMORAT    0004    000054 AMQ
    C004    000107 BETTER    0004    000102 BIAS      0003 R  000027 CIN       0005   000001 CONLV     0004    000112 CORLEN
    C003 R  000030 CTOT      0003 R  000041 DCON      0004    000053 DELT      0003 R 000073 DISS      0004    000073 DWFAC
    C004    000061 ELIMTH    0004    000052 EPS       0003 R  000050 GEN       0004   000074 GRACIM    0003 R  000047 GREF
    C000 I  000001 I         0003 I  000023 IDADJ     0003 I  000024 INDEX     0004   000056 INDXVL    0000    000005 INJPS
    C000 I  000002 J         0003 I  000000 JUNK      0000 I  000003 K         0004   000051 KROOT     0003 I  000020 LINK
    C004    000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR     0004 I 000032 LR        0004    000006 LSKEW
    C003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0004 I 000005 LV        0004    000002 LVRIN
    C005 R  000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I 000000 MQ        0004 I  000011 MXAR
    C003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO     0003 I 000043 NSYMB     0003 I  000020 NTB
    C003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT     0003 R 000045 OCIN      0004    000055 ODCON
    C003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R 000036 OW        0005 R  000010 PACCEL
    C003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R 000043 PPASS     0003 R  000042 PORAT
    C004    000071 PQRATH    0003 R  000026 PRIRCM    0003 R  000031 PROP      0003 R 000044 PST       0004    000065 SBLTH
    C004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004   000112 SPCOR     0003 R  000032 SPFAC
    C004    000072 SPMVTH    0000 R  000000 SUM       0005    000004 TRBND     0005   000005 TRCHI     0005    000006 URXBND
    C005    000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R 000037 VOLIN     0004    000101 VOLLIM
    C003 R  000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R 000034 W         0003 R  000033 WADJ
    C004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004   000067 WFAC      0004    000105 WSIM
    C004    000050 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00010(      1*          SUBROUTINE ACOM(A,B,C)                                                    000031
00010(      2*          INCLUDE MISH                                                              000031
00013(      3*          REAL SUM,A(MQ,MQ),B(MQ,MQ),C(MQ,MQ)                                      000031
00013(      4*          DO 13 I=1,MQ                                                              000031
00013(      5*          DO 13 J=1,I                                                               000102
00014(      6*          SUM=0.                                                                    000102
00014(      7*          DO 12 K=1,MQ                                                              000164
00014(      8*       12 SUM=SUM+B(I,K)*C(K,J)+C(I,K)*B(K,J)                                       000104
```

```
00146      9*          A(I,J)=SUM
00147     10*       13 A(J,I)=SUM
00152     11*          RETURN
00153     12*          END

      END OF COMPILATION:        1 DIAGNOSTICS.
```

000113
000114
000126
000171

```
@FOR,S ADJUST,ADJUST     ,ADJUST
FOR SOE3-04/18/78-01:27:36 (0,)

   SUBROUTINE ADJUST    ENTRY POINT 002333

   STORAGE USED: CODE(1) 002406; DATA(0) 000354; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS   002017
    C004    MISC   000113
    C005    STPAR  000016
    C006    CLUSTR 000131
    C007    SPPAR  000023
    C010    JOINPR 000004

   EXTERNAL REFERENCES (BLOCK, NAME)

    C011    GET
    C012    TR
    C013    DOTSQ
    C014    AMSQ
    C015    SQMTX
    C016    MINV
    C017    UNIF
    C020    CLPR
    C021    TRIMTX
    C022    DENCAL
    C023    SPLIT
    C024    FREE
    C025    CLDUMP
    C026    SEPER
    C027    SUBLIM
    C030    ELIM
    C031    CORECT
    C032    JOIN
    C033    APRIOR
    C034    SQRT
    C035    ALOG
    C036    NPRI$
    C037    NIO2$
    C040    NIO1$
    C041    EXP
    C042    XPRI
    C043    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000576 108L      0001    000606 109L      0001    000723 117L      0001    000132 134L      0001    001215 139L
    C001    001266 141L      0001    001504 200L      0001    000311 203G      0001    001565 204L      0001    001605 205L
    C001    001657 211L      0001    002055 213L      0001    000357 217G      0001    002121 250L      0001    001516 30L
    C001    000644 304G      0001    000713 320G      0001    000751 327G      0000    000137 353F      0000    000235 354F
    C001    002264 399L      0001    001750 513G      0001    002010 527G      0001    002142 555G      0001    002147 562G
```

```
0000   000251  612F      0000   000214  639F      0000   000115  701F      0000   000162  771F      0000   000174  772F
0003 R 000020  ALINK     0000 R 000071  ALP       0000 R 000072  ALPO      0004   000064  AMM       0004   000075  AMOFAC
0004   000077  AMOMAX    0004   000076  AMOMIN    0004   000100  AMORAT    0004 R 000054  AMQ       0014 R 000000  AMSQ
0033 R 000000  APRIOR    0006   000123  AREA      0004 R 000107  BETTER    0004   000102  BIAS      0006 I 000115  BUFPIX
0006 I 000116  BUFTOT    0000 R 000061  CHW       0003 R 000027  CIN       0000 R 000070  CINV      0006 I 000002  CLSNAM
0005   000001  CONLV     0004   000111  CORLEN    0003 R 000030  CIOT      0000 R 000065  CVOL      0000 R 000055  DALP
0007   000013  DAMP      0003 R 000041  DCON      0000 R 000063  DCORR     0007   000001  DELMET    0004   000053  DELT
0008 R 000067  DEN       0007   000015  DIAG      0003 R 000043  DISS      0007   000014  DOBPMS    0013 R 000000  DOTSQ
0006   000113  DUMP      0000 R 000031  DW        0004 R 000073  DWFAC     0004 R 000061  ELIMTH    0004   000052  EPS
0000 R 000062  EXP       0007   000010  EXMAX     0007   000006  EXMNSQ    0000 R 000000  FV        0007   000011  GAMLEN
0007   000000  GAMMET    0003 R 000050  GEN       0011 I 000000  GET       0004   000074  GRACTM    0003 R 000047  GREF
0000 I 000047  I         0006   000000  IBEGIN    0003 I 000023  IDADJ     0000 I 000111  IM        0003 I 000024  INDEX
0004   000066  INDXVL    0000   000305  INJPS     0006   000003  IPT       0006   000130  IQ1       0007 I 000022  ITER
0007   000020  ITERMX    0000 I 000064  ITX       0000 I 000112  J         0032 I 000000  JOIN      0003 I 000000  JUNK
0000 I 000106  K         0000 I 000032  KADTY     0000 I 000101  KC        0000 I 000104  KCC       0000 I 000102  KDC
0000 I 000022  KF        0000 I 000060  KK        0000 I 000021  KL        0006   000107  KLBC      0000 I 000113  KMAX
0004   000051  KROOT     0000 I 000025  LA        0000 I 000051  LAX       0000 I 000026  LB        0006   000127  LBUF
0006   000121  LBUFD     0000 I 000027  LD        0003 I 000020  LINK      0004   000003  LKURT     0006   000102  LNCAT
0004   000007  LOSUM     0004   000004  LOVAR     0004 I 000002  LR        0004   000036  LSKEW     0003 I 000021  LSUBS
0004   000005  LSUM      0003 I 000022  LSUPER    0004 I 000005  LV        0000 I 000030  LVA       0004   000002  LVRIN
0000 I 000050  LVX       0005 R 000012  MACCEL    0006   000122  MAXBF     0004 I 000031  MM        0004   000110  MODE
0004 I 000000  MQ        0000 I 000024  MQP       0000 I 000023  MQS       0004 I 000011  MXAR      0000 I 000020  NADJ
0003   000014  NARL      0006   000117  NBUFSD    0000 I 000077  NDSG      0000 I 000100  NDTAU     0000 I 000076  NDUM
0006   000120  NDUMP     0004   000010  NINCLS    0006   000106  NIT       0000 I 000114  NJO       0010 I 000003  NOELIM
0006   000004  NOFLD     0010 I 000002  NOJO      0006   000126  NPTS      0004 I 000070  NPTSO     0000 I 000074  NSGSQ
0003 I 000043  NSYMB     0000 I 000075  NTAUSQ    0003 I 000020  NTB       0003   000016  NTBSZM    0003   000015  NTOP
0003   000017  NWANT     0006   000125  NWDRS     0006   000124  NWDS      0007   000033  OBCOV     0007   000005  OBKURT
0007   000004  OBSKEW    0003 R 000045  OCIN      0004 R 000055  OOCON     0003 R 000047  ODEN      0003 R 000046  OPRIOR
0003 R 000035  OPROP     0000 R 000044  OVOL      0003 R 000036  OW        0005 R 000010  PACCEL    0003 R 000050  PCOND
0003 R 000025  PCUM      0000 R 000073  PFAC      0004 R 000103  PJOIN     0000 R 000054  PMOT      0003 R 000043  PPASS
0003 R 000042  PQRAT     0004 R 000071  PQRATH    0003 R 000026  PRIRCM    0000 R 000056  PRK       0006 I 000103  PRNT
0003 R 000031  PROP      0006 I 000111  PROUT     0006 I 000110  PRTME     0003 R 000044  PST       0010 R 000001  RLIM
0000 R 000105  RMIN      0000 R 000110  RR        0000 R 000043  RVOL      0004 R 000065  SBLTH     0006 I 000114  SCRAM1
0004 R 000062  SEPTH     0007   000002  SGTMET    0007   000007  SHRMIN    0000 R 000035  SK        0005 R 000002  SKBND
0005 R 000003  SKCHI     0000 R 000041  SKTES     0000 R 000052  SMOT      0000 R 000103  SPBND     0004 R 000112  SPCOR
0003 R 000032  SPFAC     0004 R 000072  SPMVTH    0007   000021  SPRED     0000 R 000033  STW       0000 R 000010  SV
0006 I 000005  SYM       0007   000016  TIMO      0007   000017  TIM1      0000 R 000056  TMOT      0006 I 000112  TOTPIX
0006 I 000001  TOTWRD    0012 R 000000  TR        0005 R 000004  TRBND     0005 R 000005  TRCHI     0000 R 000034  TRK
0000 R 000040  TRKTES    0007   000012  TSQINI    0017 R 000000  UNIF      0000 R 000036  URK       0005 R 000006  URKBND
0005 R 000007  URKCHI    0000 R 000042  URKTES    0005 R 000014  VACCEL    0004   000063  VFAC      0000 R 000053  VMOT
0003 R 000037  VOLIN     0004 R 000101  VOLLIM    0003 R 000040  VOLRT     0003 R 000050  VRIN      0004 R 000104  VRJOIN
0003 R 000034  W         0003 R 000033  WADJ      0004   000060  WADJIN    0005 R 000000  WAIT      0000 R 000037  WAITF
0004 R 000106  WDELSM    0010 R 000000  WDJOIN    0004   000067  WFAC      0000 R 000046  WINFC     0000 R 000057  WKP
0000 R 000045  WR        0004 R 000105  WSIM      0004   000050  WYINIT    0000 R 000107  WW        0004   000056  XOVFLO
0004   000057  XUNFLO
```

```
00101     1*          SUBROUTINE ADJUST(KLIN,SUM,SKEW,KURT,OSUM,OVAR)                    000013
00102     2*          INCLUDE MISH                                                       000013
00131     3*          INCLUDE CMBK10                                                     000013
00146     4*          COMMON /SPPAR/ GAMMET,DELMET,SGTMET,OBCOV,OBSKEW,OBKURT,EXMNSQ,    000013
00146     5*         1      SHRMIN,EXMAX,GAMLEN,TSQINI,DAMP,DOBPMS,DIAG,TIMO,TIM1,ITERMX,  000013
00146     6*         2      SPRED,ITER                                                  000013
00143     7*          REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)                        000013
00142     8*          REAL FV(MQMAX),SV(MQMAX)                                           000013
```

```
00143      9*          COMMON /JOINPR/WDJOIN,RLIM,NOJO,NOELIM                                    000013
00144     10*          DATA NADJ/70C/                                                            000013
00144     11*    C  PURPOSE--TO MAKE MISC ADJUSTMENTS TO THE NODE CORRESPONDING TO ILASS         000013
00144     12*    C               (1)  CHANGES CONTINUOUS STATISTICS                              000013
00144     13*    C               (2)  TESTS FOR AND INITIATES DESCRETE TRANSFORMATIONS           000013
00144     14*    C                                                                               000013
00144     15*    C                                                                               000013
00144     16*    C  ADJUST WAS DESIGNED TO INCLUDE A NUMERIC EXTRAPOLATION SYSTEM THAT           000013
00144     17*    C    HAS NOT BEEN COMPLETED.  PARAMETERS  VACCEL,  MACCEL  AND PACCEL           000013
00144     18*    C    HAVE BEEN SET TO 0 TO NULLIFY THIS SYSTEM.                                 000013
00144     19*    C                                                                               000013
00144     20*    C  VALUES  SMOT,  VMOT,  PMOT  ARE CALC. BUT NOT USED                          000013
00144     21*    C                                                                               000013
00144     22*    C                                                                         CH    000013
00144     23*    C                                                                         CH    000013
00144     24*    C  INDEX IS A FLAG THAT IS NEGATIVE IF THERE HAS BEEN A COMPLETE PASS     CH    000013
00144     25*    C     THROUGH DATA IN STATIS WITHOUT CALLING ADJUST                       CH    000013
00144     26*    C  LINK   - SIBLING NODE                                                  CH    000013
00144     27*    C  LSUB   - CHILD NODE                                                    CH    000013
00144     28*    C  LSUPER - PARENT NODE                                                   CH    000013
00144     29*    C  OVAR   - OLD COVAR MATRIX MATRIX                                       CH    000013
00144     30*    C  OVOL   - VOLUME MEASURE   MATRIX                                       CH    000013
00144     31*    C  PMOT   - CHANGE IN PRIOR SINCE LAST CALL TO ADJUST                     CH    000013
00144     32*    C  SMOT   - CHANGE IN MEAN VECTOR                                         CH    000013
00144     33*    C  SKTES -- MEASURE OF SKEWNESS                                           CH    000013
00144     34*    C  STW -STANDARDIZED WEIGHT                                               CH    000013
00144     35*    C  TRKTES - MEASURE OF KURTOSIS                                           CH    000013
00144     36*    C  URKTES - MEASURE OF KURTOSIS                                           CH    000013
00144     37*    C  VMOT   - CHANGE IN VARIANCE                                            CH    000013
00144     38*    C  VRIN   - INVERSE OF COVAR MATRIX                                       CH    000013
00144     39*    C  WAITF  - AGE FACTOR, DELAYS DECISIONS UNTIL A SIGNIFICANT NUMBER OF    CH    000013
00144     40*    C    ITERATIONS HAVE OCCURRED                                             CH    000013
00144     41*    C                                                                         CH    000013
00144     42*    C                                                                               000013
00146     43*          KL=KLIN                                                                   000013
00147     44*          KF=LSUPER(KL)                                                             000015
00147     45*    C                                                                         CH    000015
00147     46*    C  GET WORKING STORAGE FOR SUBROUTINE ADJUST (MO=NO. CHANNELS)            CH    000015
00150     47*          MOS=MO*MO                                                                 000020
00150     48*    C  CHANGE RE:PASSBACH 3/21/77                                                   000020
00151     49*          MOP=MO + 1                                                                000023
00152     50*          LA=GET(MOS)                                                               000026
00153     51*          LB=GET(MOS)                                                               000032
00154     52*          LD=GET(MOS)                                                               000036
00155     53*          LVA=GET(MOS)                                                              000042
00155     54*    C                                                                         CH    000042
00155     55*    C  CALC DIFFERENCE IN THE WEIGHT FOR CLUSTER KL (CURRENT - OLD)           CH    000042
00156     56*          DW=W(KL)-OW(KL)                                                           000046
00157     57*          KADTY=1                                                                   000051
00157     58*    C                                                                         CH    000051
00157     59*    C  INDEX IS A FLAG THAT IS NEGATIVE IF THERE HAS BEEN A COMPLETE PASS     CH    000051
00157     60*    C     THROUGH DATA IN STATIS WITHOUT CALLING ADJUST                       CH    000051
00160     61*          IF(INDEX(KL).LT.0) KADTY=2                                                000054
00160     62*    C  CALCULATE STATISTICS.                                                        000054
00160     63*    C PROPERLY,KURT SHOULD BE ADJUSTED FOR THE DISCRETE POINT                       000054
00160     64*    C EFFECT (SIMILAR TO SHEPHERD,S CORRECTION).  THIS HAS NOT YET BEEN             000054
00160     65*    C DONE, BUT SHOULD NOT HAVE ANY MAJOR EFFECT, SINCE  KURT IS USED               000054
```

```
00160   C ONLY IN THE CRUDE SCAN.                                                      000054
00162         STW=W(KL)/DW                                                             000062
00162   C  SKIP CALC IF THERE ARE SUBS FOR THIS CLUSTER                             NEW000062
00162   C                                                                              000062
00162   C  PARAMETERS FOR SPLIT TEST                                                   000062
00163         TRK=TR(KURT(KL+1),VRIN(KL+1))*STW                                        000066
00164         SK=DOTSQ(SKEW(KL+1),VRIN(KL+1))*STW                                      000102
00165         URK=(AMSQ(KURT(KL+1),VRIN(KL+1))*STW*STW-TRK*TRK/AMQ)*DW                 000116
00166         TRK=(TRK-AMQ*(AMQ+2.))*SQRT(DW)                                          000136
00166   C                                                                              000136
00166   C  DELAY FACTOR TO GIVE YOUNG CLASSES TIME TO GROW                              000136
00167         WAITF=1.+WAIT/DW                                                         000147
00167   C                                                                              000147
00167   C  ACTUAL TEST VALUES.  CHI PARAMETERS  ARE CHI**2 VALUES CALC. IN             000147
00167   C  CLINIT SIMILARLY FOR BND PARAMETERS.                                        000147
00170         TRKTES=TRK**2-DW*TRBND-TRCHI*WAITF                                       000153
00171         SKTES=SK-SKBND*DW-SKCHI*WAITF                                            000162
00172         URKTES=URK-URKBND*DW-URKCHI*WAITF                                        000171
00172   C                                                                              000171
00172   C  EXTRAPOLATE THE PARAMETERS.                                                 000171
00172   C                                                                              000171
00172   C  PREPARE VARIANCE AND VOLUME                                                 000171
00173   102   CONTINUE                                                                 000200
00174         CALL SQMTX(ALINK(LB),VRIN(KL+1))                                         000200
00175         CALL SQMTX(ALINK(LA),OVAR(KL+1))                                         000211
00176         CALL MINV(ALINK(LD),ALINK(LVA),ALINK(LB),RVOL)                           000224
00177         OVOL=-ALOG(ABS(RVOL))/AMQ-ALOG(ABS(W(KL)))                               000242
00200         WR=DW/OW(KL)                                                             000260
00201         WINFC=4.*DW*OW(KL)/W(KL)**2                                              000263
00201   C                                                                              000263
00201   C  SV = CHANGE IN CUM                                                          000263
00201   C  FV = EXTRAPOLATION FACTOR FOR SUM                                           000263
00201   C  EXTRAPOLATION DIFFERS FOR TWO TYPES OF STATISTICS (SGN(INDEX(RL))           000263
00202         DO 103 I=1,MQ                                                            000311
00205         SV(I)=SUM(I+KL)-OSUM(I+KL)                                               000311
00206   103   FV(I)=SV(I)-WR*OSUM(I+KL)                                                000313
00210         LVX=LD                                                                   000320
00211         LAX=LA                                                                   000322
00212         IF(KADTY.EQ.1) GO TO 134                                                 000324
00214         LVX=LA                                                                   000327
00215         LAX=LD                                                                   000330
00215   C                                                                              000330
00215   C  CHANGE SUME OF SQUARES                                                      000330
00215   C  EXTR SUM OF SQUARES                                                         000330
00216   134   DO 104 I=1,MQS                                                           000332
00221         ALINK(LB+I-1)=ALINK(LVX+I-1)-ALINK(LAX+I-1)                              000357
00222   104   ALINK(LVA+I-1)=ALINK(LB+I-1)-WR*ALINK(LAX+I-1)                           000361
00222   C                                                                   CH         000361
00222   C  CALCULATE SMOT (CHG IN MEAN VECTOR), VMOT (CHG IN VARIANCE),     CH         000361
00222   C  PMOT (CHG IN PRIOR) SINCE LAST CALL TO ADJUST                     CH         000361
00222   C                                                                              000361
00222   C  THESE ARE INTENDED TO INDICATE RATE OF MOTION OF CLUSTER STATISTICS          000361
00222   C  FOR INCOMPLETE SYSTEM TO CALC. NEXT ADJUSTMENT POINT.                       000361
00224         SMOT=DOTSQ(FV,VRIN(KL+1))*W(KL)/DW**2                                    000366
00225         VMOT=AMSQ(ALINK(LVA),VRIN(KL+1))*(W(KL)/DW)**2                           000401
00226         PMOT=DALP**2                                                             000420
00226   C                                                                              000420
```

```
C
C  TRACE 1--ADJUST SUMMARY PRINTOUT
      TMOT = 0.
      PRINT 701,INDEX(KL),W(KL),OW(KL),SPFAC(KL),PMOT,TMOT,SMOT,VMOT
  701 FORMAT(' ADJUST',I4,'  WEIGHT',E11.5,'  WAS',E11.5,
     1  '   SPFAC',F10.5,'   CHANGE',2E11.5,E11.5,E11.5)
      PRINT 353, TRK,SK,URK,TRKTES,SKTES,URKTES
  353 FORMAT(' STATISTICS: TRACE',E11.5,'  SKEW',E11.5,'  KURT',E11.5,
     1    7X,'TESTS (SPLIT>0): ',E11.5,6X,E11.5,6X,E11.5)
C
C  TRACE 2--ACTUAL ADJUST PRINT ON SELECTIVE
      IF(W(KL).GT.UNIF(4500.)*PPOP(KL).OR.W(KL).LE.0.OR.DW.LE.0.OR.RVOL
     * .LE.0)CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
      IF(W(KL).LE.0.OR.DW.LE.0.OR.RVOL.LE.0)PRINT 771,KL,W(KL),DW,RVOL
  771 FORMAT(1H0,'W/OVOL ERROR IN ADJUST:KL,W,NEW W,VOL',I6,3E15.7)
C
C  NADJ = ADJUSTMENT CONSTANT
      NADJ=NADJ+1
      WKP=W(KL)
      W(KL)=DW
C
C  STATISTICS--NEW WEIGHT
      KK=LSUBS(KL)
C
C
C  ADJUST TOTAL WEIGHT IN SUBCLUSTERS, IF ANY
C
C  LOCATE RIGHT-MOST NODE                                          CH
                                                                   CH
      IF(KK.EQ.0) GO TO 109
      CHW=W(KL)-WKP
  108 CTOT(KK)=CTOT(KK)+CHW
      KK=LINK(KK)
      IF(KK.NE.0) GO TO 108
  109 WR=W(KL)/DW
C
C  WR= EXTRAPOLATE MEAN
C  EXF = TEMP. EXTRAPOLATION FACTOR
C  CHANGE RE:RASSBACH 3/21/77
      EXF=WINFC*VACCEL(KADTY)
      DO 113 I=1,MC
      SUM(KL+I)=WR*(SV(I)+EXF*FV(I))
  113 OSUM(KL+I)=SUM(KL+I)
C CHANGE RE:RASSBACH 3/21/77
C
C  SET NEXT ADJUSTMENT POINTS WITH MINIMUM
      WADJ(KL)=W(KL)*(1.+DWFAC)
      IF(W(KL).LT.WSIM) WADJ(KL)=2.*W(KL)+WDELSM
C
C  DISCRETE POINT (SHEPHARD,S) CORRECTION (TO COVARIANCE ONLY)
      DCORR=(DW+WADJ(KL))/24
      IF(KADTY.EQ.2)DCORR=DW/12.
      DO 118 I=1,MQS,MQP
  118 ALINK(LB+I-1)=ALINK(LB+I-1) + DCORR
C
C  EXTRAPOLATE COVARIANCE
      EXF=WINFC*MACCEL(KADTY)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      ITX=0
  117 DO 114 I=1,MCS
  114 ALINK(LD+I-1)=WR*(ALINK(LB+I-1)+EXF*ALINK(LVA+I-1))
      CALL TRIMTX(OVAR(KL+1),ALINK(LD))
      CALL MINV(ALINK(LVA),ALINK(LD),ALINK(LD),VOLIN(KL))
      EXF =EXF*.3
C
C  EXTRAPOLATED COVARIANCE MUST BE POSITIVE DEFINITE, ELSE LOOP
      ITX=ITX+1
      IF(ITX.EQ.25)EXF=0.
      IF(VOLIN(KL).LT.0..AND.ITX.LT.26) GO TO 117
      CVOL=ALOG(ABS(VOLIN(KL)))/AMQ-ALOG(ABS(W(KL)))
C
C  ALSO REQUIRE NOT TOO RAPID CHANGE IN VOLUME
      IF(ABS(OVOL-CVOL).GT.VOLLIM.AND.ITX.LT.26) GO TO 117
C CHANGE RE:RASSBACH 3/21/77
C
C  ERROR MESSAGE
      IF(ITX.GE.10)PRINT 772,ITX,INDEX(KL),VOLIN(KL),OVOL,CVOL
  772 FORMAT(1H0,'***EXTRAPOLATION PROBLEM IN ADJUST:ITER,INDEX(KL),VOLI
     * N,OVOL','CVOL',I4,I6,3E15.7)
C CHANGE RE:RASSBACH 3/21/77
C
C  STORE COVARIANCE MATRIX
C  EFFECTIVELY DISABLE PRINT STATEMENT
C     IF (ITX .GE. 27) CALL CLPR (KL,NADJJ,SUM,SKEW,KURT)
      CALL TRIMTX(VRIN(KL+1),ALINK(LVA))
C
C  PROPORATION CALC)/EXTRAPOLATION
      EXF=WINFC*PACCEL(KADTY)
      PRK=PROP(KL)
      DEN=W(KF)-CTOT(KL)
      CINV=CIN(KL)-OCIN(KL)
      PROP(KL)=(CINV)/(DEN-ODEN(KL))
C  ERROR MESSAGE--PROPORATION CALCULATION
      IF(PROP(KL).GT.0..AND.PROP(KL).LE.1.) GO TO 139
      PRINT 639,INDEX(KL),PRK,PROP(KL),PRIRCH(KF),W(KL),CIN(KL),
     1    OCIN(KL),CINV,W(KF),CTOT(KL),DEN,ODEN(KL)
  639 FORMAT('0ALPHA ERROR:PRK,P,CM,W',I3,4E9.4/'    (ERROR CONT) CIN',
     1  3E9.4,'  W(KF),CTOT,DEN,ODEN',4E9.4)
C  CHANGE RE:RASSBACH 3/21/77
C  PRINTOUT CLUSTER
      CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
      PROP(KL)=PRK
  139 ALP=ALOG(PROP(KL))
      ALPO=ALOG(OPROP(KL))
      DALP=ALP-ALPO
      PFAC=EXP(EXF*DALP)
      PROP(KL)=PROP(KL)*PFAC
      OPROP(KL)=PROP(KL)
      PRIRCM(KF)=PRIRCM(KF)+PPOP(KL)-PRK
      CIN(KL)=CINV*PFAC
      OCIN(KL)=CIN(KL)
      ODEN(KL)=(CIN(KL)/PROP(KL)*PRIRCM(KF))
      CTOT(KL)=W(KF)-ODEN(KL)
C  ADJUST PROPORATIONS OF SUBS
      KK=LSUBS(KF)
```

```
0042A     237*        141 CALL DENCAL(KK,1./PRIRCM(KF),W(KF))                                    001266
0042E     238*            KK=LINK(KK)                                                            001300
0042E     239*            IF(KK.NE.0) GO TO 141                                                  001303
0042E     240*      C                                                                            001303
0042E     241*      C                                                                            001303
0042E     242*      C  ACTUAL TEST FOR SPLITTING                                                 001303
0042E     243*      C  SPLIT THE CLUSTER                                                         001303
0042E     244*      C                                                                            001303
0042E     245*      C  IF THERE ARE SUBS, SKIP POSSIBLE SPLITTING OF CLUSTER.                CH  001303
0042E     246*      C  CALLS ADJUST WHEN WEIGHT FOR A GIVEN CLUSTER EXCEEDS THE THRESHOLD    CH  001303
0043C     247*            IF(LSUBS(KL).NE.0) GO TO 200                                           001305
0043C     248*      C                                                                            001305
0043C     249*      C  IF TRKTES, URKTES (MEASURES OF KURTOSIS) OR                           CH  001305
0043C     250*      C    SKTES (MEASURES OF SKEWNESS) IS POSITIVE SPLIT CLUSTER              CH  001305
0043Z     251*            IF(TRKTES.LT.0..AND.SKTES.LT.0..AND.URKTES.LT.0.) GO TO 200            001314
0043Z     252*      C                                                                            001314
0043Z     253*      C                                                                            001314
0043Z     254*      C  GET WORKING STORAGE FOR SPLIT, CALL SPLIT, PRINT RESULTS AND          CH  001314
0043Z     255*      C    FREE WORKING STORAGE                                                CH  001314
0043Z     256*            NSGSQ=GET(MQS)                                                         001337
0043Z     257*            NTAUSQ=GET(MQS)                                                        001343
0043E     258*            NDUM=GET(MQS)                                                          001350
0043Z     259*            NDSG=GET(MQS)                                                          001355
0044C     260*            NDTAU=GET(MQS)                                                         001362
0044C     261*      C                                                                            001362
0044C     262*      C  DO SPLIT                                                                  001362
0044J     263*            CALL SPLIT(KL,SUM,SKEW,KURT,OSUM,OVAR,                                 001367
0044J     264*           1 LINK(LA),LINK(LB),LINK(LD),LINK(LVA),LINK(NSGSQ),                     001367
0044J     265*           2 LINK(NTAUSQ),LINK(NDUM),LINK(NDSG),LINK(NDTAU))                       001367
0044Z     266*            KC=LSUBS(KL)                                                           001434
0044Z     267*            KDC=LINK(KC)                                                           001437
0044Z     268*      C                                                                            001437
0044Z     269*      C  TRACE 4                                                                   001437
0044Z     270*            PRINT 354, INDEX(KL),W(KL),INDEX(KC),INDEX(KDC),ITER                   001441
0045J     271*      354   FORMAT(' ***HAVE SPLIT',I3,' WEIGHT',E11.5,' SUBS',2I3,' ITER',I4)     001453
0045J     272*      C                                                                            001453
0045J     273*      C  FREE STORAGE                                                              001453
0045Z     274*            CALL FREE(NSGSQ,MQS)                                                   001453
0045E     275*            CALL FREE(NTAUSQ,MQS)                                                  001457
0045E     276*            CALL FREE(NDUM,MQS)                                                    001463
0045J     277*            CALL FREE(NDSG,MQS)                                                    001467
0046C     278*            CALL FREE(NDTAU,MQS)                                                   001473
0046J     279*            CALL CLDUMP(KL)                                                        001477
0046Z     280*            GO TO 204                                                              001502
0046Z     281*      C                                                                            001502
0046Z     282*      C  DO NOT CHECK FOR SEPARATION OR TO ELIMINATE SUBS                          001502
0046Z     283*      C   ELIMINATE THIS CLUSTER IN FAVOR OF ITS SUBCLUSTERS, IF IT IS             001502
0046Z     284*      C      SPLIT WITH ODDS GREATER THAN SEPTH.                                   001502
0046J     285*        200 IF(SPFAC(KL).LE.SEPTH*SPCOR) GO TO 30                                  001504
0046E     286*            CALL SEPER(KL)                                                         001511
0046E     287*      C                                                                            001511
0046E     288*      C  DO NOT PROCESS DELETED CLUSTER FURTHER                                    001511
0046E     289*            GO TO 399                                                              001514
0046E     290*      C                                                                            001514
0046E     291*      C  ELIMINATE THE SUBS, IF IT HAS SUBS AND EITHER                             001514
0046E     292*      C     (1)  SPLITTING LESS THAT SUBLIM THRESHOLD, OR                          001514
0046E     293*      C     (2)  IT IS SIMILAR TO SUBS AND SPLITTING LESS THAN SPMVTH              001514
```

```
      30 CONTINUE
C---ELIMINATE THE SUBCLUSTERS IF THEY ARE DOMINATED BY THE MAIN
C        CLUSTER.
         SPBND=(SPFAC(KL)-OPRIOR(KL))/SPCOR
         IF((SPBND.LE.SBLTH.OR.PORAT(KL)/DW.LT.PORATH.AND.SPBND
     1     .LT.SPMVTH).AND.LSUBS(KL).NE.0) CALL SUBLIM(KL)
C     ELIMINATE THIS CLUSTER (AND PERHAPS ITS COCLUSTER) IF ITS
C        PROPORTION BECOMES TOO SMALL.
C                                                                         CH
C     ELIMINATE IF PROPORTION TOO SMALL AND ELIN PARAMETER NOELIM IS OFF
  204 IF(PROP(KL).GE.ELIMTH.OR.NOELIM.NE.0)  GO TO 205
         CALL ELIM(KL)
C                                                                         CH
C     DO NOT TRY TO PROCESS FURTHER
         GO TO 399
  205    KCC=LSUPER(KL)
         KDC=LINK(KL)
C                                                                         CH
C     CALL JOIN IF A SIMILAR CLUSTER HAS BEEN FOUND
C
C     CLUSTER MUST BE SELECTIVELY CHOSEN
C        JOIN CONTROL PARAMETER NOJO MUST BE OFF, AND THERE MUST BE A
C        STRUCTURALLY VOID JOIN AVAILABLE
C
C FIND LIKELY OVERLAPS OF THIS CLUSTER WITH THOSE FURTHER DOWN THE LIST
         IF(W(KL).LT.UNIF(WDJOIN).OR.NOJO.NE.0.OR.LINK(KL).EQ.0.
     1     OR.LSUBS(KCC).EQ.KL.AND.LINK(KDC).EQ.0) GO TO 250
         RMIN=1.E26
         K=LINK(KL)
C                                                                         CH
C     DO NOT CHECK ON RANDOM BASIS CONTROLLED BY PARAMETER PJOIN.  THIS IS
C     NECESSARY TO AVOID REPEATING BAD JOIN TRIES.
  211 IF(PJOIN.LT.UNIF(1.)) GO TO 213
         WW=W(K)/W(KL)
         CALL CORECT(FV,SUM(KL+1),WW,SUM(K+1))
         RR=(DOTSQ(FV,VRIN(KL+1))+DOTSQ(FV,VRIN(K+1))*WW)*.5/W(KL)
C                                                                         CH
C     CHECK DIFFERENCE IN DIAGONAL COVARIANCE MATRIX ELEMENTS.
C
C      ***  WARNING ***  THIS CHECK IS NOT INVARIANT UNDER GENERAL LINEAR
C                        TRANSFORMATION.
         DO 212 I=1,MO
         IM=MXAR(I+1)
C
C     THIS ERROR MIGHT OCCUR DUE TO ROUNDING ERROR.
         IF(VRIN(KL+IM)*VRIN(K+IM).LE.0.) PRINT 612,I,IM,INDEX(KL),INDEX
     1     (K),KL,K,(J,VRIN(KL+J),VRIN(K+J),J=1,MM)
  612 FORMAT('DLOG ERROR IN ADJUST:   I,IM,KL,K/VRIN=',2I5,2I3,2I7/(I5
     1     ,2E13.6))
  212 RR=RR+VRJOIN*(ALOG(ABS(VRIN(KL+IM)/VRIN(K+IM))+1.E-25)**2)
         RR=RR/(.18*(WW-1./WW)**2+1.)
C                                                                         CH
C     THE FOLLOWING VALUES SYMMETRICALLY WEIGHTED MEASURE
         IF(RR.GT.RMIN) GO TO 213
C
C     BEST SO FAR, CHECK IF CLUSTER IS JOINABLE
         KMAX=K
```

```
00544            RMIN=RR                                                                         002052
00545        213 K=LINK(K)                                                                       002055
00545      C                                                                                     002055
00545      C  LOOP OVER CLUSTERS                                                                 002055
00546            IF(K.GT.0) GO TO 211                                                            002057
00546      C                                                                                     002057
00546      C  IS BEST GOOD ENOUGH                                                                002057
00550            IF(RMIN.GT.RLIM*AMQ) GO TO 250                                                  002061
00550      C                                                                                     002061
00550      C  DO JOIN                                                                            002061
00552            NJO=JOIN(KL,KMAX,SUM,SKEW,KURT,OSUM,OVAR,LINK(LA),LINK(LB),                     002072
00552         1     LINK(LD),LINK(LVA))                                                          002072
00553        250 CONTINUE                                                                        002121
00553      C                                                                            CH       002121
00553      C                                                                            CH       002121
00553      C  CALCULATE SCALAR MEASUREMENTS OF SKEWNESS AND KURTOSIS TO TBE USED        CH       002121
00553      C     IN A TEST OF NORMALITY                                                 CH       002121
00553      C                                                                                     002121
00554                                                                                            002121
00554      C                                                                                     002121
00554      C  ZERO OUT SKEWNESS AND KURTOSIS (ACCUMULATED ONLY 1 BLOCK AT A TIME)                 002121
00554            DO 161 I=1,MQ                                                                   002121
00557        161 SKEW(KL+I)=0.                                                                   002142
00557      C                                                                                     002142
00557      C  DO RANGES OVER WHOLE TRIANGULAR ARRAY                                              002142
00561            DO 162 I=1,MM                                                                   002147
00564        162 KURT(KL+I)=0.                                                                   002147
00564      C                                                                                     002147
00564      C  ADJUST OW(56) AND SPLITTING VARIABLES                                              002147
00564      C  SPLITTING OF PARENT CLUSTER IS TO BE BOUNDED                                       002147
00566            SPFAC(KF)=AMAX1(SPFAC(KF),AMIN1(SPFAC(KF)+DW*BETTER,OPRIOR(KF)))                002150
00567            PORAT(KL)=0.                                                                    002164
00570            SPFAC(KL)=-9999.9                                                               002165
00571            IF(LSUBS(KL).NE.0) SPFAC(KL)=APRIOR(KL)                                         002167
00573            OPRIOR(KL)=SPFAC(KL)                                                            002200
00573      C                                                                                     002200
00573      C  VOLUME AND COEFFICIENT CALCULATIONS                                                002200
00574            DCON(KL)=ODCON                                                                  002203
00575            VOLIN(KL)=ABS(VOLIN(KL))*.8756510763E-26*(6.283185307/W(KL))**MQ                002205
00576            VOLRT(KL)=SQRT(VOLIN(KL))                                                       002220
00577            OW(KL)=W(KL)                                                                    002226
00577      C                                                                                     002226
00577      C  CHECK STATISTICS TYPE                                                              002226
00600            WADJ(KL)=W(KL)*(1.+DWFAC)                                                       002230
00601            INDEX(KL) = IABS(INDEX(KL))                                                  NEW002234
00602            IF(W(KL).LT.WSUM) WADJ(KL)=2.*W(KL)+WDELSM                                      002237
00604            IF (IDADJ(KL) .LE. NPTSO)  INDEX(KL) = -IABS(INDEX(KL))                         002250
00606            IDADJ(KL)=NPTSO+TOTPIX                                                          002257
00607        399 CONTINUE                                                                        002264
00607      C                                                                                     002264
00607      C  CALC WRAP-AROUND POINT                                                             002264
00607      C                                                                                     002264
00607      C  FREE ALL THE WORKING STORAGE FOR ADJUST                                   CH       002264
00610            CALL FREE(LA,MQS)                                                               002264
00611            CALL FREE(LB,MQS)                                                               002267
00612            CALL FREE(LD,MQS)                                                               002273
00613            CALL FREE(LVA,MQS)                                                              002277
```

```
00614    408*          RETURN                                                 002303
00615    409*          END                                                    002405

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S ALFREE,ALFREE,ALFREE
FOR SOE3-04/18/78-01:27:45 (0,0)

   SUBROUTINE ALFREE    ENTRY POINT 000032

   STORAGE USED: CODE(1) 000043; DATA(0) 000007; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    FREE
    C007    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000007 IOL        0003 R 000020 ALINK     0004    000064 AMM      0004    000075 AMOFAC   0004    000077 AMOMAX
    C004    000076 AMOMIN     0004   000100 AMORAT    0004    000054 AMQ      0004    000107 BETTER   0004    000102 BIAS
    C003  P 000027 CIN        0005   000001 CONLV     0004    000111 CORLEN   0003 R 000030 CTOT     0003 R  000041 DCON
    C004    000053 DELT       0003 R 000043 DISS      0004    000073 DWFAC    0004    000061 ELIMTH   0004    000052 EPS
    C003  P 000050 GEN        0004   000074 GRACTM    0003 R  000047 GREF     0003 I  000023 IDADJ    0003 I  000024 INDEX
    C004    000066 INDXVL     0000   000002 INJP$     0003 I  000000 JUNK     0000 I  000000 KL       0000 I  000001 KLK
    C004    000051 KROOT      0003 I 000020 LINK      0004    000003 LKURT    0004    000007 LOSUM    0004    000004 LOVAR
    C004  I 000002 LR         0004   000006 LSKEW     0003 I  000021 LSUBS    0004    000005 LSUM     0003 I  000022 LSUPER
    C004  I 000005 LV         0004   000002 LVRIN     0005 R  000012 MACCEL   0004    000001 MM       0004    000110 MODE
    C004    000000 MQ         0004 I 000011 MXAR      0003    000014 NARL     0004    000010 NINCLS   0004    000070 NPTSO
    C003  I 000043 NSYMB      0003 I 000020 NTB       0003    000016 NTBSZM   0003    000015 NTOP     0003    000017 NWANT
    C003  R 000045 OCIN       0004   000055 ODCON     0003 R  000047 ODEN     0003 R  000046 OPRIOR   0003 R  000035 OPROP
    C003  R 000036 OW         0005 R 000010 PACCEL    0003 R  000050 PCOND    0003 R  000025 PCUM     0004    000103 PJOIN
    C003  R 000043 PPASS      0003 R 000042 PQRAT     0004    000071 PQRATH   0003 R  000026 PRIRCM   0003 R  000031 PROP
    C003  R 000044 PST        0004   000065 SBLTH     0004    000062 SEPTH    0005    000002 SKBND    0005    000003 SKCHI
    C004    000112 SPCOR      0003 R 000032 SPFAC     0004    000072 SPMNTH   0005    000004 TRBND    0005    000005 TRCHI
    C005    000006 URKBND     0005   000007 URKCHI    0005 R  000014 VACCEL   0004    000063 VFAC     0003 R  000037 VOLIN
    C004    000101 VOLLIM     0003 R 000040 VOLRT     0003 R  000050 VRIN     0004    000104 VRJOIN   0003 R  000034 W
    C003  R 000033 WADJ       0004   000060 WADJIN    0005    000000 WAIT     0004    000106 WDELSM   0004    000067 WFAC
    C004    000105 WSIM       0004   000050 WTINIT    0004    000056 XOVFLO   0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*            SUBROUTINE ALFREE(KLHED,LEN)                                          000000
00101       2*        C  THIS ROUTINE FREES THE STRING STARTED BY KLHED.                        000000
00101       3*            INCLUDE MISH                                                          000000
00131       4*            IF(KLHED.EQ.0) RETURN                                                 000000
00133       5*            KL=KLHED                                                              000004
00134       6*         10 KLK=LINK(KL)                                                          000007
00136       7*            CALL FREE(KL,LEN)                                                     000011
00136       8*            KL=KLK                                                                000015
```

```
00137      9*            IF(KL)10,99,10                                  000017
00142     10*        99 KLHEQ=0                                         000020
00143     11*           RETURN                                          000021
00144     12*           END                                             000042

      END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S AMSQ,AMSQ,AMSQ
FOR SOE3-C4/18/78-01:27:48 (0,0)

   FUNCTION AMSQ        ENTRY POINT 000250

   STORAGE USED: CODE(1) 000272; DATA(0) 000047; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003   CLUS   002017
    [004   MISC   000113
    [005   STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006   NERR3$

   STORAGE ASSIGNMENT. (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001    000142 12L       0001    000015 135G      0001    000205 14L       0001    000034 1406      0001    000047 1476
    [001    000111 1636      0001    000154 1776      0003 R  000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    [004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ       0000 R  000000 AMSG
    [000 P  000001 AMSQD     0004    000107 BETTER    0004    000102 BIAS      0003 R  000027 CIN       0000 R  000005 COL
    [005    000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT      0003 R  000041 DCON      0004    000053 DELT
    [003 R  000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS       0003 R  000050 GEN
    [004    000074 GPACTM    0003 R  000047 GREF      0000 I  000002 I         0003 I  000023 IDADJ     0000 I  000006 IKLOC
    [003 I  000024 INDEX     0004    000066 INDXVL    0000    000013 INJP$     0000 I  000012 IP        0000 I  000003 J
    [000 I  000011 JP        0003 I  000000 JUNK      0000 I  000010 K         0000 I  000007 KJLOC     0004    000051 KR$OT
    [003 I  000020 LINK      0004    000003 LKURT     0004    000007 LOSUB     0004    000004 LOVAR     0004 I  000002 LR
    [004    000006 LSKEW     0003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV
    [004    000002 LVRIN     0005 R  000012 MACCEL    0004    000001 MM        0004    000116 MODE      0004 I  000000 MQ
    [004 I  000011 MXAR      0003     000014 NARL     0004    000010 NINCLS    0004    000070 NPTSO     0003 I  000043 NSYMB
    [003 I  000020 NTB       0003     000016 NTBSZM   0003     000015 NTOP      0003     000017 NWANT    0003 R  000045 OCIN
    [004    000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW
    [005 R  000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS
    [003 P  000042 PQRAT     0004    000071 PQRATH    0003 R  000026 PRIRCM    0003 R  000031 PROP      0003 R  000044 PST
    [000 R  000004 ROW       0004    000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI
    [004    000112 SPCOR     0003 R  000032 SPFAC     0004    000072 SPMVTH    0005    000004 TRBND     0005    000005 TRCHI
    [005    000006 URKBND    0005     000007 URKCHI   0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN
    [004    000101 VOLLIM    0003 R  000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R  000034 W
    [003 R  000033 WADJ      0004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC
    [004    000105 WSIM      0004    000050 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

00001[  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*            REAL FUNCTION AMSQ(AM,AMET)                                          000010
00101       2*      C       CALCULATES THE TRACE OF THE SQUARE OF THE MATRIX AM, RELATIVE       000010
00101       3*      C         TO THE METRIC AMET.                                                000010
00101       4*      C       AMSQ = TRACE(AM*AMET*AM*AMET)                                        000010
00102       5*            INCLUDE MISH                                                         000010
00131       6*            REAL AM(475),AMET(475)                                               000010
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00132      7*            AMSQ=0.                                                    000010
00133      8*            AMSQD=0.                                                   000011
00134      9*            DO 20 I=1,MQ                                               000015
00137     10*            DO 19 J=1,I                                                000034
00142     11*            ROW=0.                                                     000037
00143     12*            COL=0.                                                     000040
00144     13*            IKLOC=MXAR(I)                                              000041
00145     14*            KJLOC=MXAR(J)                                              000043
00146     15*            DO 10 K=1,J                                                000047
00146     16*      C   D      CALL IXCHEK(1,I,J,K,IKLOC+1,KJLOC+1 )                  000047
00151     17*            ROW=ROW+AM(IKLOC+1)*AMET(KJLOC+1)                          000056
00152     18*            COL=COL+AM(KJLOC+1)*AMET(IKLOC+1)                          000062
00153     19*            IKLOC=IKLOC+1                                              000066
00154     20*         10 KJLOC=KJLOC+I                                              000071
00156     21*            KJLOC=KJLOC+J                                              000075
00157     22*            IF(I.EQ.J) GO TO 12                                        000077
00161     23*            JP=J+1                                                     000102
00162     24*            DO 11 K=JP,I                                               000105
00163     25*      C   D      CALL IXCHEK(2,I,J,K,IKLOC+1,KJLOC )                    000105
00165     26*            ROW=ROW+AM(IKLOC+I)*AMET(KJLOC)                            000121
00166     27*            COL=COL+AM(KJLOC)*AMET(IKLOC+1)                            000125
00167     28*            IKLOC=IKLOC+1                                              000131
00170     29*         11 KJLOC=KJLOC+K                                              000134
00172     30*         12 IF(I.EQ.MQ) GO TO 18                                       000142
00174     31*            IKLOC=IKLOC+I                                              000143
00175     32*            IP=I+1                                                     000146
00176     33*            DO 13 K=IP,MQ                                              000150
00176     34*      C   D      CALL IXCHEK(3,I,J,K,IKLOC,KJLOC )                      000150
00201     35*            ROW=ROW+AM(IKLOC)*AMET(KJLOC)                              000164
00202     36*            COL=COL+AM(KJLOC)*AMET(IKLOC)                              000170
00203     37*            IKLOC=IKLOC+K                                              000174
00204     38*         13 KJLOC=KJLOC+K                                              000177
00206     39*         14 CONTINUE                                                   000205
00207     40*         19 AMSQ=AMSQ+ROW*COL                                          000205
00211     41*         20 AMSQD=AMSQD+ROW*COL                                        000212
00213     42*            AMSQ=AMSQ+AMSQ-AMSQD                                       000220
00213     43*      C       WE MUST COUNT EACH OFF-DIAGONAL TWICE. AMSQD AVOIDS DOUBLE-000220
00214     44*      C          COUNTING THE DIAGONAL TERMS.                           000220
00214     45*            RETURN                                                     000223
00215     46*            END                                                        000271

   END OF COMPILATION:          1 DIAGNOSTICS.
```


ORIGINAL PAGE IS OF POOR QUALITY

```
@FOR,S APRIOR,APRIOR,APRIOR
FOR SOE3-04/18/78-01:27:52 (0,0)

   FUNCTION APRIOR       ENTRY POINT 000013

   STORAGE USED: CODE(1) 000015; DATA(0) 000005; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C003 R 000020 ALINK      0004   000064 AMM        0004   000075 AMOFAC    0004   000077 AMOMAX    0004   000076 AMOMIN
    C004   000100 AMORAT     0004 R 000054 AMQ        0000 R 000000 APRIOR     0004   000107 BETTER    0004 R 000102 BIAS
    C003 R 000027 CIN        0005   000001 CONLV      0004   000111 CORLEN     0003 R 00003C CTOT      0003 R 000041 DCON
    C004   000053 DELT       0003 R 000043 DISS       0004   000073 DWFAC     0004   000051 ELIMTH    0004   000052 EPS
    C003 R 000050 GEN        0004   000074 GRACTM     0003 R 000047 GREF      0003 I 000023 IDADJ     0003 I 000024 INDEX
    C004   000066 INDXVL     0000   000001 INJP$      0003 I 000000 JUNK      0004   000051 KROOT     0003 I 000020 LINK
    C004   000003 LKURT      0004   000007 LOSUM      0004   000004 LOVAR     0004 I 000002 LR        0004   000006 LSKEW
    C003 I 000021 LSUBS      0004   000005 LSUM       0003 I 000022 LSUPER    0004 I 000005 LV        0004   000002 LVRIN
    C005 R 000012 MACCEL     0004   000001 MM         0004   000110 MODE      0004   000000 MQ        0004 I 000011 MXAR
    C003   000014 NARL       0004   000010 NINCLS     0004   000070 NPTSO     0003 I 000043 NSYMB     0003 I 000020 NTB
    C003   000016 NTBSZM     0003   000015 NTOP       0003   000017 NWANT     0003 R 000045 OCIN      0004   000055 ODCON
    C003 R 000047 ODEN       0003 R 000046 OPRIOR     0003 R 000035 OPROP     0003 R 000036 OW        0005 R 000010 PACCEL
    C003 R 000050 PCOND      0003 R 000025 PCUM       0004   000103 PJOIN     0003 R 000043 PPASS     0003 R 000042 PQRAT
    C004   000071 PQRATH     0003 R 000026 PRIRCM     0003 R 000031 PROP      0003 R 000044 PST       0004   000065 SBLTH
    C004   000062 SEPTH      0005   000002 SKBND      0005   000003 SKCHI     0004   000112 SPCOR     0003 R 000032 SPFAC
    C004   000072 SPMVIH     0005   000004 TRBND      0005   000005 TRCHI     0005   000006 URKBND    0005   000007 URKCHI
    C005 R 000014 VACCEL     0004 R 000063 VFAC       0003 R 000037 VOLIN     0004   000101 VOLLIM    0003 R 000040 VOLRT
    C003 R 000050 VRIN       0004   000104 VRJOIN     0003 R 000034 W         0003 R 000053 WADJ      0004   000060 WADJIN
    C005   000000 WAIT       0004   000106 WDELSM     0004   000067 WFAC      0004   000135 WSIM      0004   000050 WTINIT
    C004   000056 XOVFLO     0004   000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            FUNCTION APRIOR(KL)                                                        000000
00101      2*      C  THIS ROUTINE CALCULATES THE APRIORI PROBABILITY FOR THE                       000000
00101      3*      C     CLUSTER KL AS OPPOSED TO ITS TWO SUBCLUSTERS KA AND KB.                   000000
00101      4*      C     THE PROBABILITY CALCULATED HAS NOTHING TO DO WITH THE                     000000
00101      5*      C     DATA, BUT CONTAINS ONLY THE USERS' BIAS IN FAVOR OF FEWER                 000000
00101      6*      C     CLUSTERS.  IF APRIOR IS SET TOO LARGE (WHERE 1. IS TOO LARGE)             000000
00101      7*      C     THEN THE ALGORITHM WILL GENERATE TOO MANY CLUSTERS (I.E.:                  000000
00101      8*      C     ONE CLUSTER PER DATA POINT)  EXTREMELY SMALL VALUES OF                     000000
00101      9*      C     APRIOR WILL DECREASE THE NUMBER OF CLUSTERS CREATED.  IN GENERAL,          000000
```

```
00101     10*      C      EXCEPT FOR EXTREMELY STATISTICALLY SENSITIVE PROBLEMS,                  000000
00101     11*      C      ANY SMALL VALUE OF APRIOR IS SUFFICIENT; IN THE LIMIT OF                000000
00101     12*      C      INFINITE DATA, THE ALGORITHM WILL FIND THE CLUSTERS ANYHOW.             000000
00101     13*      C      APRIOR MUST BE POSITIVE, TYPICALLY 3.**(-MO).                           000000
00103     14*           INCLUDE MISH                                                            000000
00131     15*           APRIOR=VFAC*AMO+BIAS                                                    000000
00132     16*           RETURN                                                                  000003
00133     17*           END                                                                     000014

      END OF COMPILATION:          1 DIAGNOSTICS.
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S CBLO,CBLO,CBLO
FOR SOE3-04/18/78-01:27:55 (0,0)

   BLOCK DATA

   STORAGE USED: CODE(1) 000000; DATA(0) 000000; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    020504
    C004    MISC    000113
    C005    STPAR   000016
    C006    SPPAR   000023
    C007    INITL   000003
    C010    JOINPR  000004

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C003  R 000020 ALINK      0004    000064 AMM       0004  R 000075 AMOFAC    0004  R 000077 AMOMAX    0004  R 000076 AMOMIN
    C004  R 000100 AMORAT     0004    000054 AMQ       0004  R 000107 BETTER    0004  R 000102 BIAS      0007  R 000002 CHANIN
    C003  R 000027 CIN        0005  R 000001 CONLV     0004  R 000111 CORLEN    0003  R 000030 CTOT      0006  R 000013 DAMP
    C003  R 000041 DCON       0006  R 000001 DELMET    0004    000053 DELT      0007  R 000001 DEVINI    0006  L 000015 DIAG
    C003  R 000043 DISS       0006  R 000014 DOBPMS    0004  R 000073 DWFAC     0004  R 000061 ELIMTH    0004    000052 EPS
    C006  R 000010 EXMAX      0006  R 000006 EXMNSQ    0006  R 000013 GAMCEN    0006  R 000000 GAMMET    0003  R 000050 GEN
    C004  R 000074 GRACTM     0003  R 000047 GREF      0003  I 000023 IDADJ     0003  I 000024 INDEX     0004  I 000066 INDXVL
    C006    000022 ITER       0006  I 000020 ITERMX    0003  I 000000 JUNK      0004    000051 KROOT     0003  R 000020 LENG
    C003  I 000020 LINK       0004    000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR     0004  R 000002 LR
    C004    000006 LSKEW      0003  I 000021 LSUBS     0004    000005 LSUM      0003  I 000022 LSUPER    0004    000005 LV
    C004    000002 LVRIN      0005  R 000012 MACCEL    0004    000001 MM        0004  I 000110 MODE      0004    000000 MQ
    C004  I 000011 MXAR       0003  I 000014 NARL      0004    000010 NINCLS    0010  I 000003 NOELIM    0010    000002 NOJO
    C004    000070 NPTSQ      0003  I 000004 NSYMB     0003  I 000020 NTB       0003  I 000016 NTBSZM    000     000015 NTOP
    C003    000017 NWANT      0006  R 000000 OBCOV     0006  R 000005 OBKURT    0006  R 000004 OBSKEW    000     000045 OCIN
    C004    000055 ODCON      0003  R 000047 ODEN      0003  R 000046 OPRIOR    0003  R 000035 OPROP     0003  R 000036 OW
    C005  R 000010 PACCEL     0003  R 000050 PCOND     0003  R 000025 PCUM      0004  R 000103 PJOIN     0003  R 000043 PPASS
    C003  R 000042 PQRAT      0004  R 000071 PQRATH    0003  R 000026 PRIRCM    0003  R 000031 PROP      0003  R 000044 PST
    C010  R 000001 RLIM       0004  R 000065 SBLTH     0004  R 000062 SEPTH     0006  R 000002 SGTMET    0006  R 000007 SHRMIN
    C005  P 000002 SKBND      0005    000003 SKCHI     0004  R 000112 SPCOR     0003  R 000032 SPFAC     0004  R 000072 SPMVTH
    C006  R 000021 SPRED      0006  R 000016 TIMD      0006  R 000017 TIM1      0005  R 000004 TRBND     0005    000005 TRCHI
    C006    000012 TSQINI     0005  R 000006 URKBND    0005    000007 URKCHI    0005  R 000014 VACCEL    0004  R 000063 VFAC
    C003  R 000037 VOLIN      0004  R 000101 VOLLIM    0003  R 000040 VOLRT     0003  R 000050 VRIN      0004  R 000104 VRJOIN
    C003  P 000034 W          0003  R 000033 WADJ      0004  R 000060 WADJIN    0005  R 000000 WAIT      0004  R 000106 WDELSM
    C010  R 000000 WDJOIN     0004  R 000067 WFAC      0004  R 000105 WSIM      0004  R 000050 WTINIT    0007  R 000000 WTNEW
    C004  R 000056 XOVFLO     0004  R 000057 XUNFLO

0000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*           BLOCK DATA                                                                  000000
00102      2*           INCLUDE MISH                                                                000000
00102      3*      C  CORE DEFINITION                                                               000000
0013(      4*           PARAMETER NAREA = 8500                                                 NEW000000
00131      5*           REAL LENG(NAREA)                                                       -01000000
00132      6*           EQUIVALENCE (LINK(1),LENG(1))                                               000000
00132      7*      C  ALLOCATION SETUP                                                              000000
00133      8*           PARAMETER NARLEN=NAREA-4                                                    000000
```

```
      PARAMETER NTOPI=NTBSIZ+87
C THE CONSTANT SHOULD NOT HAVE TO BE LARGER THAN 1.
      PARAMETER NTBSOM=NTBSIZ-1
      DATA MODE /1/
C  CALLOC
      DATA NARL,NTOP,NTBSZM/NARLEN,NTOPI,NTBSOM/
C  STATIS
      DATA XOVFLO,XUNFLO /8.,-8./
C
C  INIT CLUSTER
      DATA INDXVL/1/
C  ADJUST
      DATA SB…TH,ELIMTH,SEPTH,PQRATH,SPMVTH/-50.,.5E-2,9.,.005,7./
      DATA BE?TER /.16/
C     D      DATA PACCEL,MACCEL,VACCEL,VOLLIM/2*.35,2*.35,2*.29,.3/
      DATA PACCEL,MACCEL,VACCEL,VOLLIM/6*0.,.3/
      DATA DWFAC,GRACTM,AMOFAC,AMOMIN,AMOMAX,AMORAT/1.6,4.4,.5,.8,4.,
     1  1.5/
      DATA WSIM,WDELSM /137.03602,8./
      DATA PJOIN/.47,VRJOIN/.3/
      DATA WAIT,CONLV,SKBND,TRBND,URKBND/80.,2.1,0.,0.,0./
C  STATIS
      DATA CORLEN,SPCOR /0.,1./
C  SPLIT
      COMMON/SPPAR/ GAMMET,DELMET,SGTMET,OBCOV,OBSKEW,OBKURT,EXMNSQ,
     1  SHRMIN,EXMAX,GAMCEN,TSQINI,DAMP,DOBPMS,DIAG,TIM0,TIM1,ITERMX,
     2     SPRED,ITER
      LOGICAL DIAG
      DATA DIAG/.FALSE./
      DATA OBCOV,OBSKEW,OBKURT/.7,.3,1./
      DATA GAMMET,DELMET,SGTMET,GAMCEN/.4,1.6,1.,.3/
      DATA SHRMIN,EXMNSQ,EXMAX/.04,.04,8./
      DATA TIM0,TIM1,DOBPMS,DAMP,ITERMX/10.,8.,1E-3,.25,150/
      DATA SPRED,VFAC,BIAS,WFAC/.11,-2.,-1.,9./
C  PRIOR
      COMMON /INITL/WTNEW,DEVINI,CHANIN
C  CLINIT
      DATA WTNEW,DEVINI,WTINIT,CHANIN,WADJIN/8.,100.,.001,40.,40./
C  JOIN
      COMMON /JOINPR/WDJOIN,RLIM,NOJO,NOELIM
      DATA WDJOIN,RLIM,NOJO,NOELIM/200.,.5,0,0/
      END
```

END OF COMPILATION: 1 DIAGNOSTICS.

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S CLASY,CLASY,CLASY
FOR SOE3-04/18/78-01:27:58 (2,0)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000036; DATA(0) 000004; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    INFORM 000407
    C004    CLUSTR 000131
    C005    CLUS   002017
    C006    MISC   000113
    C007    STPAR  000016
    C010    ARRAY  037201

   EXTERNAL REFERENCES (BLOCK, NAME)

    C011    SETUP9
    C012    READTP
    C013    MULTI
    C014    CLUSMP
    C015    NINTR$
    C016    NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C005 P 000020 ALINK    0006   000064 AMM      0006   000075 AMOFAC   0006   000077 AMOMAX   0006   000076 AMOMIN
    C006   000100 AMORAT   0006   000054 AMQ      0004   000123 AREA     0010 I 000000 ARRAY    0003   000351 BESTVC
    C006   000107 BETTER   0006   000102 BIAS     0004 I 000115 BUFPIX   0004 I 000116 BUFTOT   0005 R 000027 CIN
    C004 I 000002 CLSNAM   0007   000001 CONLV    0006   000111 CORLEN   0005 R 000030 CTOT     0003   000053 DATAPE
    C005 R 000041 DCON     0006   000053 DELT     0005 R 000043 DISS     0003   000112 DIVSIZ   0004   000113 DUMP
    C003   000062 DUPSYM   0006   000073 DWFAC    0006   000061 ELIMTH   0006   000052 EPS      0003   000217 FETVC2
    C003   000161 FETVEC   0003   000071 FILESV   0005 R 000050 GEN      0006   000074 GRACTM   0005 R 000047 GREF
    C003   000000 HEAD     0003   000255 HISVEC   0004   000000 IBEGIN   0003   000123 IBLOCK   0005 I 000023 IDADJ
    C005 I 000020 IDATA    0005 I 000024 INDEX    0006   000066 INDXVL   0003   000313 INVERT   0004 I 000003 IPT
    C004   000130 IQ1      0005 I 000000 JUNK     0003   000113 KEEPLV   0004   000107 KLBC     0006   000051 KROOT
    C000 I 000001 LAST     0004   000127 LBUF     0004   000121 LBUFD    0005 I 000020 LINK     0006   000003 LKURT
    C004 I 000102 LNCAT    0006   000007 LOSUM    0006   000004 LOVAR    0006 I 000002 LR       0006   000006 LSKEW
    C005 I 000021 LSUBS    0006   000005 LSUM     0005 I 000022 LSUPER   0006 I 000005 LV       0006   000002 LVRIN
    C007 R 000012 MACCEL   0003   000052 MAPTAP   0004   000122 MAXBF    0003   000072 MAXCLS   0003   000064 MAXDIV
    C003   000055 MAXFET   0003   000074 MAXFLD   0003   000065 MINDIV   0006   000001 MM       0006   000110 MODE
    C006   000000 MQ       0006 I 000011 MXAR     0005   000014 NARL     0004   000117 NBUFSD   0004   000120 NDUMP
    C006   000010 NINCLS   0004   000106 NIT      0003   000073 NOCLS2   0003   000121 NOCLS3   0003   000100 NOFEAT
    C003   000101 NOFET2   0003   000102 NOFET4   0004   000004 NOFLD    0003   000075 NOFLD2   0003   000076 NOFLD3
    C003   000111 NOGRP    0003   000110 NOHIST   0003   000107 NOSPEC   0003   000077 NOTRFD   0004   000126 NPTS
    C006   000070 NPTSO    0005 I 000043 NSYMB    0005 I 000020 NTB      0005   000016 NTBSZM   0005   000015 NTOP
    C005   000017 NWANT    0004   000125 NWDRS    0004   000124 NWDS     0005 R 000045 OCIN     0006   000055 ODCON
    C005 P 000047 ODEN     0005 R 000046 OPRIOR   0005 R 000035 OPROP    0005 R 000036 OW       0007 R 000010 PACCEL
    C003   000056 PAGSIZ   0005 R 000050 PCOND    0003   000122 PCTSZ    0005 R 000025 PCUM     0006   000103 PJOIN
    C005 R 000043 PPASS    0005 R 000042 PQRAT    0006   000071 PQRATH   0005 R 000026 PRIRCM   0004 I 000103 PRNT
    C005 R 000031 PROP     0004 I 000111 PROUT    0003   000114 PRTLEV   0004 I 000110 PRTME    0005 R 000044 PST
    C003   000054 SAVTAP   0006   000065 SBLTH    0004 I 000114 SCRAM1   0006   000062 SEPTH    0003   000067 SERIAL
    C007   000002 SKBND    0007   000003 SKCHI    0003   000120 SPCBAS   0006   000112 SPCOR    0005 R 000032 SPFAC
    C003   000066 SPLMAX   0006   000072 SPMVTH   0004 I 000005 SYM      0005   000057 TAPCHK   0003   000070 TAPESV
```

ORIGINAL PAGE IS
OF POOR QUALITY

0003	000063	THRSYM	0010	I	037200	TOP	0000	I	000000	TOPID	0004	I	000112	TOTPIX	0004	I	000001	TOTWRD
0007	000004	TRBND	0007		000005	TRCHI	0003		000060	TRNSYM	0003		000061	TSTSYM	0007		000006	URKBND
0007	000007	URKCHI	0007	R	000014	VACCEL	0003		000103	VARSIZ	0003		000104	VARSZ2	0003		000105	VARSZ4
0006	000063	VFAC	0005	R	000037	VOLIN	0006		000101	VOLLIM	0005	R	000040	VOLRT	0005	R	000050	VRIN
0006	000104	VRJOIN	0005	R	000034	W	0005	R	000033	WADJ	0006		000060	WADJIN	0007		000000	WAIT
0006	000106	WDELSM	0006		000067	WFAC	0006		000105	WSIM	0006		000050	WTINIT	0003		000116	XHGH
0003	000117	XLOW	0006		000056	XOVFLO	0003		000106	XSIZ	0006		000057	XUNFLO	0003		000115	YSIZ

```
0000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
0010(      1*      C*************************************************************************** 000000
0010(      2*      C*************************************************************************** 000000
0010(      3*      C*                                                                          000000
0010(      4*      C                                                                           000000
0010(      5*      C                                                                           000000
0010(      6*      C                                                                           000000
0010(      7*      C  ACOM    MULT MATRIX B BY MATRIX C AND STORE RESULT IN MATRIX A.       1  000000
0010(      8*      C                                                                       2  000000
0010(      9*      C  ADJUST  CHECK TO SEE IF ANY OF THE CLASSES ARE POTENTIALLY TWO       3  000000
0010(     10*      C          CLASSES AND REFER THOSE THAT SHOULD BE TWO TO 'SPLIT'.       4  000000
0010(     11*      C                                                                       5  000000
0010(     12*      C  ALFREE  FREE STRING STARTED BY KLHED.                                6  000000
0010(     13*      C                                                                       7  000000
0010(     14*      C  AMSQ    CALCULATE THE TRACE OF THE SQUARE OF THE MATRIX AM,          8  000000
0010(     15*      C          RELATIVE TO THE MATRIX AMET.                                 9  000000
0010(     16*      C                                                                      10  000000
0010(     17*      C  APRIOR  CALCULATE THE APRIORI PROBABILITY FOR THE CLUSTER KL AS     11  000000
0010(     18*      C          OPPOSED TO ITS TWO SUBCLUSTERS KA AND KB.                   12  000000
0010(     19*      C                                                                      13  000000
0010(     20*      C  BFINIT  FUNCTION TO INITIALIZE TWO BUFFERS.                         14  000000
0010(     21*      C                                                                      15  000000
0010(     22*      C  BYTRAN  MOVE BYTES FROM ONE STRING TO ANOTHER.                      16  000000
0010(     23*      C                                                                      17  000000
0010(     24*      C  CALLOC  STORAGE ALLOCATOR, GET,S AND FREE,S NODES FROM CHAIN.       18  000000
0010(     25*      C                                                                      19  000000
0010(     26*      C  CLASY   CLUSTERS DATA USING A VARIABLE CLUSTER COUNT MAXIMUM        20  000000
0010(     27*      C          LIKELIHOOD CLUSTERING ALGORITHM.                            21  000000
0010(     28*      C                                                                      22  000000
0010(     29*      C  CLASY1  READ AND SCRAMBLE ONE BUFFER OF DATA                        23  000000
0010(     30*      C                                                                      24  000000
0010(     31*      C  CLDUMP  PRINT OUT ALL THE CLASSES VIA ROUTINE 'CLPR'.               25  000000
0010(     32*      C                                                                      26  000000
0010(     33*      C  CLINIT  INITIALIZE THE CLUSTERING ALGORITHM.                        27  000000
0010(     34*      C                                                                      28  000000
0010(     35*      C  CLPR    PRINT ALL THE VARIABLES BELONGING TO SOME CLASS INDEXED BY KL 29  000000
0010(     36*      C                                                                      30  000000
0010(     37*      C  CLPRM   PRINT ALL VARIABLES BELONGING TO SOME CLASS INDEXED BY KL.  31  000000
0010(     38*      C                                                                      32  000000
0010(     39*      C  CLUSMP  PRINT THE CLUSTER MAP,REPRESENTING EACH PIXEL BY A SYMBOL   33  000000
0010(     40*      C          WHICH INDICATES THE CLUSTER TYPE.                           34  000000
0010(     41*      C                                                                      35  000000
0010(     42*      C  CLUST   TAKE THE INPUT POINT AND CLASSIFIES IT FOR THE PURPOSE OF   36  000000
0010(     43*      C          GENERATING A MAP.                                           37  000000
0010(     44*      C                                                                      38  000000
0010(     45*      C  COLINV  INVERT A GIVEN SYMMETRIC POSITIVE DEFINITE MATRIX(S) BY     39  000000
```

```
00010C     46*      C            COMPUTING A TRIANGULAR FACTORIZATION (R), INVERTING R TO          40      000000
00010C     47*      C            OBTAIN A, AND THEN FINDING THE INVERSE OF S.                      41      000000
00010C     48*      C                                                                              42      000000
00010C     49*      C     CORECT CALCULATE AN UNWEIGHTED NORMALIZED VECTOR.                        43      000000
00010C     50*      C                                                                              44      000000
00010C     51*      C     DATFIX INITIALIZE VARIABLES                                              45      000000
00010C     52*      C                                                                              46      000000
00010C     53*      C     DENCAL ADJUST THE DENOMINATOR OFFSET AND PROPORTION OF KL                47      000000
00010C     54*      C                                                                              48      000000
00010C     55*      C     DISC   CALCULATE RANDOM NUMBER                                           49      000000
00010C     56*      C                                                                              50      000000
00010C     57*      C     DOTSQ  CALCULATE THE INNER PRODUCT V . V RELATIVE TO THE METRIC AMET      51      000000
00010C     58*      C                                                                              52      000000
00010C     59*      C     EIGROT CALL SYSTEM ROUTINES TO GENERATE AN EIGENROTATION OF A            53      000000
00010C     60*      C            LP * LP SUBMATRIX OF THE ARRAY R.  THE EIGENVALUES ARE            54      000000
00010C     61*      C            RETURNED IN E AND THE EIGENVECTOR MATRIX IS IN V(DIM NM*NP),      55      000000
00010C     62*      C            WHERE THE SECOND INDEX RUNS OVER EIGENVECTORS, AND THE            56      000000
00010C     63*      C            FIRST WITHIN THEM.                                                57      000000
00010C     64*      C                                                                              58      000000
00010C     65*      C     ELIM   ELIMINATES THE GIVEN CLUSTER FROM THE CLUSTER TREE AND            59      000000
00010C     66*      C            FREES STORAGE.                                                    60      000000
00010C     67*      C                                                                              61      000000
00010C     68*      C     FDLINT RETURNS THE PIXEL NUMBERS OF THOSE PIXELS ON A GIVEN LINE         62      000000
00010C     69*      C            THAT ARE CONTAINED WITHIN THE BOUNDARIES OF A NON-RECT FIELD.     63      000000
00010C     70*      C                                                                              64      000000
00010C     71*      C     FIND   LOCATES ONE OF A GIVEN SET OF SYMBOLS IN A BUFFER.                65      000000
00010C     72*      C                                                                              66      000000
00010C     73*      C     ISPLIT JUDGES FOR CLDUMP TO MAKE MAP WHETHER A CLUSTER IS BEST           67      000000
00010C     74*      C            SPLIT OR NOT                                                      68      000000
00010C     75*      C                                                                              69      000000
00010C     76*      C     JOIN   RAISES THE HYPOTHESIS THAT KA AND KB ARE THE SAME CLUSTER         70      000000
00010C     77*      C                                                                              76      000000
00010C     78*      C     MINV   CALCULATES A = INVERSE OF C,                                      77      000000
00010C     79*      C            AND DOES THE ACTION OF JOINING.                                   71      000000
00010C     80*      C            CALCULATES THE DETERMINANT OF C IN VOL.                           78      000000
00010C     81*      C                                                                              79      000000
00010C     82*      C                                                                              83      000000
00010C     83*      C     MTVEC  CALCULATES THE PRODUCT OF THE COLUMNS OF MATRIX A AND             84      000000
00010C     84*      C            VECTOR V AND STORE THE RESULT IN VECTOR U.                        85      000000
00010C     85*      C                                                                              86      000000
00010C     86*      C     MULTI  CALLS ROUTINES TO INITIALIZE THE CLUSTERING ALGORITHM.            87      000000
00010C     87*      C                                                                              88      000000
00010C     88*      C                                                                              81      000000
00010C     89*      C     MPVS   CALCULATES TENSOR PRODUCT OF V AND C.                             82      000000
00010C     90*      C     LINERD READ AND UNPACK ONE SCAN LINE OF DATA.                            75      000000
00010C     91*      C                                                                              72      000000
00010C     92*      C     LAREAD READ A CARD AND RETURN A VALUE AS DIRECTED BY CARD COL 1.         73      000000
00010C     93*      C                                                                              74      000000
00010C     94*      C     MLT    ADD MATRIX B TO MATRIX C AND STORE RESULT IN MATRIX A.            80      000000
00010C     95*      C     MVEC   MULTIPLIES THE ROWS OF MATRIX A BY VECTOR V AND STORES THE        89      000000
00010C     96*      C            RESULT IN VECTOR U.                                               90      000000
00010C     97*      C                                                                              91      000000
00010C     98*      C     NRAND  CALCULATES A RANDOM NUMBER.                                       92      000000
00010C     99*      C                                                                              93      000000
00010C    100*      C     NUMBER CALCULATES THE COMPUTATIONAL VALUES OF A SERIES OF CHARS.         94      000000
00010C    101*      C                                                                              95      000000
00010C    102*      C     NXTCHR LOCATE THE NEXT NONBLANK SYMBOL IN A BUFFER.                      96      000000
```

```
00010(    103*      C
00010(    104*      C  READIP  CORRDINATES THE ROUTINES TO READ FIELDS OF DATA FROM THE
00010(    105*      C          IMAGE TAPE AND STORE IT ON A DRUM FILE FOR CLASY SUBROUTINES.
00010(    106*      C               (INTERNAL ROUTINES--EMPTY, BUFILL
00010(    107*      C
00010(    108*      C  SEPER   PERMANENTLY SPLITS A CLUSTER.  (THE CLUSTER HAS BEEN
00010(    109*      C          PREVIOUSLY SPLIT BY THE ROUTINE SPLIT, AND SUFFICIENT
00010(    110*      C          STATISTICS HAVE BEEN GATHERED TO CONFIRM THAT THE CLUSTER
00010(    111*      C          CAN BE SPLIT ON A STATISTICALLY SIGNIFICANT BASIS.
00010(    112*      C
00010(    113*      C  SETUP9  READ AND ANALYZE ALL CARD INPUT TO THE PROGRAM.
00010(    114*      C
00010(    115*      C  SPLIT   GUESSES THE OPTIMAL AXIS TO SPLIT THE CLUSTER KL USING
00010(    116*      C          SKEWNESS AND KURTOSIS DATA, AND GENERATES TWO CLUSTERS
00010(    117*      C          CORRESPONDING TO THE PROBABLE HALVES OF THE OLD CLUSTER.
00010(    118*      C
00010(    119*      C  SQMTX   EXPAND MATRIX AM FROM TRIANGULAR FORM AND MAKES A
00010(    120*      C          MQ * MQ SQUARE SYMMETRIC MATRIX IN SQ(DIM MQ*MQ).
00010(    121*      C
00010(    122*      C  STATIS  (1) CLASSIFY EACH INPUT POINT ON A FRACTIONAL, PROBABILISTIC
00010(    123*      C          BASIS,  (2) UPDATES THE VARIOUS STATISTICAL PARAMETERS
00010(    124*      C          ASSOCIATED WITH THE CLASSES INDICATED,  (3) CHECKS TO SEE
00010(    125*      C          IF ANY OF THESE CLASSES IS POTENTIALLY TWO, (4) REFERS ANY
00010(    126*      C          SO CLASSIFIED TO ROUTINE 'SPLIT'.
00010(    127*      C
00010(    128*      C  STOFLO
00010(    129*      C
00010(    130*      C
00010(    131*      C
00010(    132*      C  SUBLIM  ELIMINATES THE SUBCLUSTERS OF THE NODE KLHED.
00010(    133*      C
00010(    134*      C  TAPERD  READS THE MULTISPECTRAL SCANNER DATA TAPE, UNPACKS THE
00010(    135*      C          REQUESTED DATA AND RETURNS IT UNPACKED TO THE CALLING ROUTINE
00010(    136*      C
00010(    137*      C  TAPHDR  READS THE HEADER RECORD AND UNPACKS NECESSARY DATA FROM THE
00010(    138*      C          RECORD.
00010(    139*      C
00010(    140*      C  TR      CALCULATES THE TRACE OF THE MATRIX AM RELATIVE TO THE
00010(    141*      C          METRIC AMET.
00010(    142*      C
00010(    143*      C  TRFREE  FREES THE TREE HEADED BY KLHED.
00010(    144*      C
00010(    145*      C  TRIMTX  PUT THE LOWER TRIANGLE OF SQ (DIM MQ * MQ) INTO SYMMETRIC
00010(    146*      C          MATRIX FORM IN TRI.
00010(    147*      C
00010(    149*      C  UNIF    RETURNS A RANDOM NUMBER.
00010(    149*      C
00010(    150*      C  UNPAK1  UNPACKS UP TO 32-BIT IBM 360 WORDS INTO 36-BIT WORD IMAGES
00010(    151*      C          WHICH CAN BE READ BY THE UNIVAC 1108.
00010(    152*      C
00010(    153*      C  VMTV
00010(    154*      C
00010(    155*      C
00010(    156*      C  VPV     VA = VA + FAC * VB
00010(    157*      C
00010(    158*      C  WRTMTX  PRINTS THE DOUBLE-PRECISION COVARIANCE MATRICES
00010(    159*      C

     97   000000
     98   000000
     99   000000
    100   000000
    101   000000
    102   000000
    103   000000
    104   000000
    105   000000
    106   000000
    107   000000
    108   000000
    109   000000
    110   000000
    111   000000
    112   000000
    113   000000
    114   000000
    115   000000
    116   000000
    117   000000
    118   000000
    119   000000
    120   000000
    121   000000
    122   000000
    123   000000
    124   000000
    125   000000
    126   000000
    127   000000
    128   000000
    129   000000
    130   000000
    131   000000
    132   000000
    133   000000
    134   000000
    135   000000
    136   000000
    137   000000
    138   000000
    139   000000
    140   000000
    141   000000
    142   000000
    143   000000
    144   000000
    145   000000
    146   000000
    147   000000
    148   000000
    149   000000
    150   000000
    151   000000
    152   000000
    153   000000
```

```
C                                                                            154
C  XP(A) = E ** (-A/2)                                                   MRCH
C  KROTIN = ROOT CLUSTER VALUES, SCALARS ONLY, NO VECTORS OR MATRICES     MRCH
C  KROT   = KROTIN                                                        MRCH
C  BMQ    = 2/3 AMQ  (AMQ = MQ = NUMBER OF CHANNELS)                      MRCH
C  CLASY2 = BLOCK OF DATA POINTS                                          MRCH
C  NDO    = NO. OF PTS. IN BLOCK CLASY2  (SET IN SUBROUTINE CLASY2)       MRCH
C  IDO    = INTERNAL INDEX OF POINTS IN BLOCK                             MRCH
C  PCUM   = CUMULATED PROBABILITY OF SUBCLUSTERS                          MRCH
C  PRIRCM = CUMULATION OF APRIORI PROBABILITY OF SUBCLUSTERS              MRCH
C  PPASS  = 1-INITIALLY, RESET TO P (PROB. BEING PASSED DOWN THE TREE)    MRCH
C  ISEC   = 0-INITIALLY,                                                  MRCH
C  KL     = NAME USED FOR CURRENT CLUSTER                                 MRCH
C  KFATH  = FATHER OF CURRENT NODE                                        MRCH
C              + IF *CURRENT MEAN* CLUSTER  (CHANGING AS DATA PROCESSED)MRCH
C  INDEX(KL) = NAME OF CLUSTER       ABS(INDEX(KL))                       MRCH
C              - IF *FIXED MEAN* CLUSTER (1 PASS THROUGH DATA, NO CHANGEMRCH
C  DOTSQ  = FUNCTION TO CALC. A TRANS MATRIX A                            MRCH
C  WUSE   =                                                               MRCH
C  DCON   = A =(A) E**(-DCON/2)                                           MRCH
C  PRO    = PROPORTION OF CLUSTER     (A)                                 MRCH
C  CIN    = PART OF PROPORTION CAL. SYSTEM                                MRCH
C  CTOT   = PART OF PROPORTION CAL. SYSTEM                                MRCH
C  PCOND  = CONDITIONAL PROBABILITY (PROB. OF POINT GIVEN THE CLASS KL)  MRCH
C  VOLRT  = SQ. ROOT OF VOLIN                                             MRCH
C  SPUSE  = SPFAC/SPCOR   (SPLITTING FACTOR)                              MRCH
C  SPUSE .LT. XUNFLO=PARENT FAVORED, .GT. XOVFLO= OFFSPRING FAVORED       MRCH
C  SPFAC  = LOG OF PROB. RATIO BETWEEN PARENT AND OFFSPRING               MRCH
C  SPCOR  = FUDGE FACTOR (RASSBACH DETERMINATION, NOMINAL VALUE 1)        MRCH
C  XUNFLO = CONSTANTS SET IN MISC                                         MRCH
C  XOVFLO = CONSTANTS SET IN MISC                                         MRCH
C  PST    = WORKING PROBABILITY FOR CLUSTER                               MRCH
C  W      = SUM OF PROB. OF POINTS IN CLUSTER                             MRCH
C  PPASSK = LOCAL                                                         MRCH
C  P      = FINAL PROBABILITY FOR CLUSTER                                 MRCH
C  NPTSO  = TOTAL NO. OF POINTS                                           MRCH
C  PLIM   = FIXED MARGIN                                                  MRCH
C  DISC(MONTE)                                                            MRCH
C  ALOW+  = TEMP NEEDED TO CALC. ADJ. FACTORS FOR MEANS, COVAR, ETC.      MRCH
C   ALPHA   WHEN MATRIX IS ADJUSTED BY RANK 1 MATRIX                      MRCH
C  PQRAT  = MEASURE OF DISTANCE BETWEEN PARENT + SUM OF DAUGHTERS         MRCH
C           WEIGHTED SUM OF ZQS (USED BY JOIN)                            MRCH
C  ZQ     = TEMP USED FOR NORMALIZED DIFF PROB. OF PARENT AND SUM OF      MRCH
C           OFFSPRING
C  ZQS    = TEMP USED FOR SQUARE OF ZQ                                    MRCH
C  VOLIN  = DETERMINANT FACTORS   (PI, ETC.)                              MRCH
C  VOLPT  = SQ RT. OF VOLIN                                               MRCH
C  PROPL  = TEMP USED IN. PROP. CALC. SYSTEM                              MRCH
C  REL    = TEMP VECT= X - SUM/W  (POSITION OF POINT RELATIVE TO MEAN).   MRCH
C  COVEC  = TEMP VECTOR = VRIN * REL                                      MRCH
C  COFI   = TEMP, ADJUSTMENT COEFFICIENT USED FOR VRIN AND VOLIN          MRCH
C  OVAR   = OLD COVARIANCE MATRIX (NOT INVERTED)                          MRCH
C  WDISS  = WEIGHTED X ** 2  (X**2 X P)                                   MRCH
C  SKEW   = SKEWNESS (CONTRACTED - A RANK 1 TENSOR)                       MRCH
C  KURT   = KURTOSIS (CONTRACTED - A RANK 2 TENSOR)                       MRCH
C  WADJ   = W VALUE FOR THE NEXT ADJUST TEST OF THE CLUSTER               MRCH
C  IDADJ  = NPTSO VALUE FOR THE NEXT ADJUST OF THE CLUSTER                MRCH
```

```
00100 C         (LAST ADJUST + SAMPLE SIZE) ADJUST CALLED IF W GT WADJ OR   MRCH   000000
00100 C          NPTSQ GT IDADJ)                                            MRCH   000000
00100 C  KADJ   = INDEX OF CLUSTER TO BE ADJUSTED                           MRCH   000000
00100 C                                                                     MRCH   000000
00100 C                                                                     MRCH   000000
00100 C  ADJUST PURPOSE--SELECTS AND SUPERVISES ALL PERIOD TRANSFORMATIONS  MRCH   000000
00100 C                  ON THE CLUSTER                                     MRCH   000000
00100 C  MQS    = SIZE OF SQUARE ARRAY                                      MRCH   000000
00100 C  DW     = CHANGE IN WEIGHT OF CLUSTER                               MRCH   000000
00100 C  KADTY  = 1 OR 2, 1 INITIALLY, CLUSTER ADJUSTMENT TYPE              MRCH   000000
00100 C  STW    = RATIO OF CURRENT WEIGHT TO CHANGE IN WEIGHT               MRCH   000000
00100 C  TRK    = TRACE OF KURTOSIS                                         MRCH   000000
00100 C  SK     = SQUARE OF SKEWNESS                                        MRCH   000000
00100 C  URK    = TRACEFREE PART OF KURTOSIS                                MRCH   000000
00100 C  WAIT   = AGING FACTOR                                              MRCH   000000
00100 C  WAITF  = FACTOR EXPRESSING TIME IN WEIGHT PERIOD                   MRCH   000000
00100 C  TRBND  = BOUNDS ON CHI SQUARES                                     MRCH   000000
00100 C  SKBND  = BOUNDS ON CHI SQUARES                                     MRCH   000000
00100 C  URKBND = BOUNDS ON CHI SQUARES                                     MRCH   000000
00100 C  TRCHI  = CHI SQUARE VALUES CALC. IN CLINIT                         MRCH   000000
00100 C  SKCHI  = CHI SQUARE VALUES CALC. IN CLINIT                         MRCH   000000
00100 C  URKCHI = CHI SQUARE VALUES CALC. IN CLINIT                         MRCH   000000
00100 C                                                                            000000
00100 C                                                                            000000
00100 C                                                                            000000
00100 C*      THIS PROGRAM CLUSTERS THE DATA USING AN ALGORITHM (VARIABLE          000000
00100 C*      CLUSTER COUNT MAXIMUM LIKLIHOOD CLUSTERING) DEVELOPED BY MIKE        000000
00100 C*      RASSBACH OF NASA-EOD.                                                000000
00100 C*                                                                           000000
00100 C*      THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA IN EITHER THE         000000
00100 C*      LARSYS 22 OR THE UNIVERSAL FORMAT.  THE DATA TAPE SHOULD BE          000000
00100 C       ASSIGNED TO UNIT 3.  DRUM ACCESS IS BY RANDIO  RANDOM ACCESS         000000
00100 C*      ROUTINES- RREAD,RWRITE,RINIT                                         000000
00100 C*                                                                           000000
00100 C*********************************************************************      000000
00100 C*                                                                           000000
00101       IMPLICIT INTEGER (A-X)                                                 000000
00103       INCLUDE COMBK1,LIST                                                    000001
00103 COMBK1 PROC                                                                  000001
00104       COMMON /INFORM/HEAD(42), MAPTAP,    DATAPE,    SAVTAP,    MAXEET,      000001
00104      1                PAGSIZ, TAPCHK,     TRNSYM,    TSTSYM,                 000001
00104      2                DUPSYM, THRSYM,     MAXDIV,    MINDIV,    SPLMAX,      000001
00104      3                SERIAL, TAPESV,     FILESV,                            000001
00104      4                MAXCLS, NOCLS2,     HAXFLD,    NOFLD2,    NOFLD3,      000001
00104      5                NOTRFD, NOFEAT,     NOFET2,    NOFET4,    VARSIZ,      000001
00104      6                VARSZ2, VARSZ4,     XSIZ   ,   NOSPEC,    NOHIST,      000001
00104      *                NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,    YSIZ  ,        000001
00104      8                XHGH   , XLOW   ,   SPCBAS,    NOCLS3,    PCTSZ  ,     000001
00104      9          IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(30),              000001
00104      *          INVERT(30), BESTVC(30)                                       000001
00104  END COMBK1                                                                  000001
00105       INCLUDE CMBK10,LIST                                                    000001
00105 CMBK10 PROC                                                                  000001
00106       PARAMETER MAXPOP=61                                                    000001
00106 C                                                                            000001
00107       COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(MAXPOP) ,           000001
00107      1               LNCAT, PRNT(4) , KLBC , PRTME, PROUT, TOTPIX,          000001
```

```
00107    255*          2  DUMP,SCRAM1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD                                  000001
00107    255*         3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1                                    000001
00107    255*     C                                                                                   000001
00110    255*          INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT                000001
00110    255*         1 ,CLSNAM                                                                       000001
00110    255*     C                                                                                   000001
00111    255*          EQUIVALENCE ( NIT,PRNT(4) )                                                    000001
00111    255*     C                                                                                   000001
00112    255*          DATA PRNT(4)/10/                                                               000001
00112    255*     C                                                                                   000001
00112    255*     C  BUFPIX = SIZE OF 'ARRAY'/NO. OF CHANNELS                                CH       000001
00112    255*     C  CLSNAM =                                                                        000001
00112    255*     C                                                                                  000001
00112    255*     C  IBEGIN = FIRST AVAILABLE WORD ON FAST STORAGE                           CH       000001
00112    255*     C  IPT = 1ST AVAIL WORD IN SCRATCH AREA 'ARRAY'                                    000001
00112    255*     C  KLBC =                                                                          000001
00112    255*     C                                                                                  000001
00112    255*     C  LBUFD = LEGTH OF LAST RECORD IN ORIG DATA AS COPIED TO WORD AD DRUM     CH      000001
00112    255*     C  LNCAT =                                                                         000001
00112    255*     C                                                                                  000001
00112    255*     C  NBUFSD = SIZE OF ORIG DATA/SIZE OF BUFFER 'ARRAY'                       CH      000001
00112    255*     C  NDUMP = SIZE OF AVAILABLE AREA IN 'ARRAY'                                       000001
00112    255*     C  NIT = NO. ITERATIONS REQUESTED FOR SCRAMBLED DATA SET, DEFLT TO 10      CH      000001
00112    255*     C  NOFLD =                                                                         000001
00112    255*     C                                                                                  000001
00112    255*     C  PROUT = PRINT FLAG                                                              000001
00112    255*     C                                                                                  000001
00112    255*     C  PRTME = PRINT FLAG                                                              000001
00112    255*     C                                                                                  000001
00112    255*     C  SCRAM1 = 1ST WORD OF AVAIL FAST STORAGE + LENGTH OF ORIG DATA UNLESS    CH      000001
00112    255*     C       ORIG DATA MOVED THEN 1ST WORD OF AVAIL FAST STORAGE                CH      000001
00112    255*     C  SUFTOT = NO. WORDS AVAIL IN SCRATCH AREA 'ARRAY'                        CH      000001
00112    255*     C  SYM = WORD ARRAY OF SYMBOLS                                                     000001
00112    255*     C  TOTPIX = TOTWRD/MQ,  INITIALIZED IN STATUS                                      000001
00112    255*     C  TOTWRD = NO. WORDS IN ORIGINAL DATA ON DRUM                             CH      000001
00112    255*     C                                                                                  000001
00112    255*      END CMBK10                                                                        000001
00114    256*          DATA   TOP/18000/ ,  TOPID/1000/                                              000001
00118    257*     C    TOPID SHOULD EQUAL VALUE FOR NAREA IN CBLO                                    000001
00117    258*          INCLUDE  MISH,LIST                                                            000001
00117    259*     C                                                                          CH      000001
00117    260*     C                                                                        NEW000001
00117    260*     MISH  PROC                                                                         000001
00117    260*     C                                                                                  000001
00120    260*          PARAMETER NTBSIZ=32                                                           000001
00120    260*     C  NTBSIZ MUST BE A POWER OF TWO.  (NTB & NTBSIZ ARE USED IN THE                   000001
00120    260*     C       STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREE.)                        000001
00120    260*     C                                                                                  000001
00120    260*     C    THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF                               000001
00120    260*     C      SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.                           000001
00120    260*     C                                                                                  000001
00121    260*          PARAMETER NARRS=3                                                             000001
00122    260*          PARAMETER NVECS=3                                                             000001
00123    260*          PARAMETER NSCALS=25                                                           000001
00124    260*          INTEGER JUNK(12), NTB(NTBSIZ), LINK(999)                                      000001
00124    260*     C                                                                                  000001
00124    260*     C  ** TYPE STATEMENTS CANNOT BE USED FOR DIMENSION STATEMENTS ON IBM 370           000001
```

```
      REAL W(1),PORAT(1),VOLIN(1),VOLRT(1),OCON(1),PPASS(1),PCOND(1)                    000001
      INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)                             000001
      REAL VRIN(475),GEN(999),GREF(999),ALINK(1)                                        000001
      REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(1),OW(1),SPFAC(1)                       000001
      REAL PRIRCM(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)                   000001
C   THE FOLLOWING STRUCTURES THE CLUSTER NODES.  'GEN' IS                               000001
C     USED LATER AS THE BASE FOR ALL THE ARRAYS.  (SEE CLINIT)                          000001
C          'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,                         000001
C             RATHER THAN IN CALLING SEQUENCES.                                         000001
      EQUIVALENCE (LINK(31),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),                   000001
     1    INDEX(27),PCUM(26),PRIRCM(25),CIN(24),CTOT(23),                               000001
     2    PROP(22),SPFAC(21),WADJ(20),W(19),OPROP(18),OW(17),                           000001
     3    VOLIN(16),VOLRT(15),OCON(14),PORAT(13),DISS(12),PPASS(12),                    000001
     4    PST(11),OCIN(10),PCOND(7),VRIN(7),GEN(7),OPRIOR(9),ODEN(8),                   000001
     5    GREF(8),NTB(31))                                                              000001
      EQUIVALENCE (LINK(1),ALINK(1))                                                    000001
C                                                                                       000001
      COMMON/CLUS/  JUNK,NARL,NTOP,NTBSZM,NWANT,LINK                                    000001
C                                                                                       000001
C                                                                                       000001
C  WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE                              000001
C     ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV                              000001
C     FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED                             000001
C     FOR CONVENIENCE.  (FOR EXAMPLE, GEN(LSUM+KL+I) IS THE                             000001
C     VECTOR SUM(KL+I) .)                                                               000001
      INTEGER MXAR(31),LR(NARRS),LV(NVECS)                                              000001
C  ** TYPE STATEMENTS CANNOT BE USED FOR DIMENSION STATEMENTS ON IBM 370                000001
      EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),                                          000001
     1   (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)                        000001
C                                                                                       000001
      COMMON /MISC/ MQ,MM,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT,                      000001
     1   AMQ,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH,                   000001
     2   INDXVL,WFAC,NPTSO,PORATH,SPMVTH,DWFAC,GRACTM,AMOFAC,                          000001
     3   AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,                    000001
     4   BETTER,MODE,CORLEN,SPCOR                                                      000001
C                                                                                       000001
      REAL PACCEL(2),MACCEL(2),VACCEL(2)                                                000001
C                                                                                       000001
      COMMON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,                   000001
     1   PACCEL,MACCEL,VACCEL                                                          000001
C     MQMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)                           000001
C                                                                                       000001
C                                                                                       000001
      PARAMETER MQMAX = 8                                                               000001
C                                                                                       000001
C     MQTMAX=(MQMAX*(MQMAX+1))/2                                                        000001
      INTEGER GET                                                                       000001
C                                                                                       000001
      PARAMETER MQTMAX = 36                                                             000001
C                                                                                       000001
 END MISH                                                                               000001
      COMMON /ARRAY/ ARRAY(16000),TOP                                                   NEWD000001
      DIMENSION IDATA (1)                                                               NEWD000001
C                                                                                       CH  -D1000001
      EQUIVALENCE  (LINK(1),IDATA(1)     )                                              000001
C*                                                                                      000001
C*    CALL SETUP9 TO READ INPUT CARDS                                                   000001
```

```
00147     267*      C*                                                                    000001
00150     268*            CALL SETUP9                                                     000001
00151     269*            TOP = 16000                                                   NEW000003
00151     270*      C*                                                                    000003
00151     271*      C*    CALL READTP TO READ CLASS AND FIELD DEFINITION CARDS AND TO READ 000003
00151     272*      C*    THE FIELDS OF DATA FROM THE IMAGE TAPE AND TO STORE DATA ON DRUM 000003
00151     273*      C*                                                                    000003
00151     274*      C  NWDS = TOTAL NO. WORDS AVAIL ON DRUM (SEE CALL TO RINIT IN READTP)  CH 000003
00152     275*         10 CALL READTP(ARRAY,TOP,LAST,IDATA,TOPID)                         000005
00152     276*      C     SET PRINT COUNTERS                                            000005
00153     277*            PRIME=1                                                       000015
00154     278*            PROUT=1                                                       000017
00155     279*            LNCAT=0                                                       000021
00155     280*      C*                                                                  000021
00155     281*      C*    CALL MULTI TO PERFORM CLUSTERING                              000021
00155     282*      C*                                                                  000021
00156     283*            CALL MULTI (  ARRAY(IPT)  )                                   000022
00156     284*      C                                                                   CH 000022
00156     285*      C                                                                   000022
00156     286*      C  PRINT CLUSTER MAP                                                000022
00157     287*            CALL CLUSMP                                                 NEW000027
00160     288*            STOP                                                        -02000031
00161     289*            END                                                           000035

      END OF COMPILATION:       1 DIAGNOSTICS.
```

```
@FOR,S CLOUMP,CLOUMP,CLOUMP
FOR SOE3-04/18/78-01:28:07 (2,0)

   SUBROUTINE CLOUMP    ENTRY POINT 000146

   STORAGE USED: CODE(1) 000153; DATA(0) 000033; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003   CLUS   002017
    [004   MISC   000113
    [005   STPAR  000016
    [006   CLUSTR 000131

   EXTERNAL REFERENCES (BLOCK, NAME)

    [007   ISPLIT
    [010   CLPR
    [011   NPRT$
    [012   NIO2$
    [013   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001     000023 10L        0001     000121 11L        0001     000114 17L        0001     000036 19L        0000     000007 210F
    [000     000017 310F       0001     000020 9L         0001     000134 99L        0003 R   000020 ALINK      0004     000064 AMM
    [004     000075 AMOFAC     0004     000077 AMOMAX     0004     000376 AMOMIN     0004     000100 AMORAT     0004     000054 AMQ
    [006     000123 AREA       0004     000107 BETWER     0004     000102 BIAS       0006 I   000115 BUFPIX     0006 I   000116 BUFTOT
    [003 R   000027 CIN        0006 I   000002 CLSNAM     0005     000001 CONLV      0004     000111 CORLEN     0003 R   000030 CTOT
    [003 R   000041 DCON       0004     000053 DELT       0003 R   000043 DISS       0006     000113 DUMP       0004     000073 DWFAC
    [004     000061 ELIMTH     0004     000052 EPS        0003 R   000050 GEN        0004     000074 GREFTM     0003 R   000047 GREF
    [006     000000 IBEGIN     0003 I   000023 IDADJ      0000 L   000000 IJSP       0003 I   000024 INDEX      0004     000066 INDXVL
    [000     000024 INJP$      0006     000003 IPT        0006     000130 IQ1        0007 L   000000 ISPLIT     0003 I   000030 JUNK
    [000 I   000006 KEEP       0000 I   000002 KL         0006     000107 KLBC       0000 I   000004 KLIN       0004     000051 KROOT
    [000 I   000003 KROT       0006     000127 LBUF       0006     000121 LBUFD      0020 I   000005 LEVEL      0003 I   000020 LINK
    [004 I   000003 LKURT      0006     000102 LNCAT      0004     000007 LOSUM      0004     000004 LOVAR      0004 I   000002 LR
    [004 I   000006 LSKEW      0003 I   000021 LSUBS      0004 I   000005 LSUM       0003 I   000022 LSUPER     0004 I   000005 LV
    [004     000002 LVRIN      0005 R   000012 MACCEL     0006     000122 MAXBF      0004     000001 MM         0004     000110 MODE
    [004     000000 MQ         0004 I   000011 MXAR       0003     000014 NARL       0006     000117 NBUFSD     0006     000120 NBUMP
    [004     000010 NINCLS     0006     000106 NIT        0000 I   000001 NOFKL      0006     000004 NOFLO      0006     000126 NPTS
    [004     000070 NPTSO      0003 I   000043 NSYMB      0003 I   000020 NTB        0003     000016 NTBSZM     0003     000015 NTOP
    [003     000017 NWANT      0006     000125 NWORS      0006     000124 NWDS       0003 R   000045 OCIN       0004     000055 OCON
    [003 P   000047 ODEN       0003 R   000046 OPRIOR     0003 R   000035 OPROP      0003 P   000036 OW         0005 R   000010 PACCEL
    [003 R   000050 PCOND      0003 R   000025 PCUM       0004     000103 PJOIN      0003 R   000043 PPASS      0003 R   000042 PORAT
    [004     000071 PORATH     0003 R   000026 PRIRCM     0006 I   000103 PRNT       0003 R   000031 PROP       0006 I   000111 PROUT
    [006 I   000110 PRTME      0003 P   000044 PST        0004     000065 SBLTH      0006 I   000114 SCRAM1     0004     000062 SEPTH
    [005     000002 SKBND      0005     000003 SKCHI      0004     000112 SPCOR      0003 R   000032 SPFAC      0004     000072 SPMVTH
    [006 I   000005 SYM        0006 I   000112 TOTPIX     0006 I   000001 TOTWRD     0005     000004 TRBND      0005     000005 TRCHI
    [005     000006 URKBND     0005     000007 URKCHI     0005 R   000014 VACCEL     0004     000063 VFAC       0003 R   000037 VOLIN
    [004     002101 VOLLIM     0003 R   000040 VOLRT      0003 R   000050 VRIN       0004     000104 VRJOIN     0003 R   000034 W
    [003 P   000033 WADJ       0004     000060 WADJIN     0005     000000 WAIT       0004     000106 WDELSM     0004     000067 WFAC
    [004     000105 WSIM       0004     000050 WTINIT     0004     000056 XOVFLO     0004     000057 XUNFLO
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0000( *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101     1*          SUBROUTINE CLDUMP(KLHED)                                                    000000
00101     2*      C   THIS ROUTINE PRINTS OUT ALL THE CLASSES VIA ROUTINE 'CLPR'.                 000000
00103     3*          INCLUDE MISM                                                                000000
00131     4*          INCLUDE CMBK10                                                              000000
00141     5*          LOGICAL ISPLIT,IJSP                                                         000000
00141     6*          NOFKL=0                                                                     000000
00142     7*          KL=KLHED                                                                    000000
00143     8*          KROT=LSUPER(KL)                                                             000003
00144     9*          KLIN=LINK(KL)                                                               000005
00145    10*          PRINT 210,INDEX(KL),KL                                                      000007
00151    11*      210 FORMAT('1DUMP OF OBSERVED CLUSTERS FROM',I3,I7)                             000015
00152    12*          LEVEL=0                                                                     000015
00153    13*          GO TO 11                                                                    000016
00154    14*        9 LEVEL=LEVEL+1                                                               000020
00155    15*       10 NSYMB(KL)=0                                                                 000023
00156    16*          IF(ISPLIT(KL)) GO TO 19                                                     000024
00160    17*          NOFKL=NOFKL+1                                                               000030
00161    18*          NSYMB(KL)=NOFKL                                                             000034
00162    19*       19 CALL CLPR(KL,LEVEL,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                         000036
00163    20*          IJSP=ISPLIT(KL)                                                             000055
00164    21*          PRINT 310,IJSP,PROUT                                                        000061
00170    22*      310 FORMAT(1H0,' ISPLIT,PROUT= ',2I5)                                           000070
00171    23*          KEEP=KL                                                                     000070
00172    24*          KL=LSUBS(KL)                                                                000072
00173    25*          IF(KL.NE.0.AND.(ISPLIT(KEEP).OR.PROUT.LE.2)) GO TO 9                        000074
00175    26*       17 KL=LINK(KEEP)                                                               000114
00176    27*          IF(KL.EQ.KLIN) GO TO 99                                                     000116
00200    28*       11 IF(KL)10,29,10                                                              000121
00203    29*       29 KL=LSUPER(KEEP)                                                             000122
00204    30*          LEVEL=LEVEL-1                                                               000125
00205    31*          KEEP =KL                                                                    000130
00206    32*          IF(KL.NE.KROT) GO TO 17                                                     000131
00210    33*       99 RETURN                                                                      000134
00211    34*          END                                                                         000152

      END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S CLINIT,CLINIT,CLINIT
FOR SOE3-04/18/78-01:28:11 (1,0)

   SUBROUTINE CLINIT    ENTRY POINT 000445

   STORAGE USED: CODE(1) 000462; DATA(0) 000071; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003    CLUS    002017
    [004    MISC    000113
    [005    STPAR   000016
    [006    CLUSTR  000131
    [007    INITL   000003

   EXTERNAL REFERENCES (BLOCK, NAME)

    [010    GET
    [011    SQRT
    [012    XPRI
    [013    ALOG
    [014    NPRT$
    [015    NIO2$
    [016    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001    000006 143G      0001    000037 152G      0001    000076 163G      0001    000106 171G      0001    000232 225G
    [001    000261 235G      0000    000005 273F      0003 R  000020 ALINK     0004 R  000064 AMM       0004    000075 AMOFAC
    [004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004 R  000054 AMQ       0006    000123 AREA
    [004    000107 BETTER    0004    000102 BIAS      0006 I  000115 BUFPIX    0006 I  000116 BUFTOT    0007 R  000002 CHANIN
    [003 R  000027 CIN       0006 I  000002 CLSNAM    0005 R  000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT
    [003 R  000041 DCON      0004    000053 DELT      0007 R  000001 DEVINI    0000 R  000003 DEV2WT    0003 R  000043 DISS
    [006    000113 DUMP      0004    000073 DWFAC     0004    000051 ELIMTH    0004    000052 EPS       0003 R  000050 GEN
    [010 I  000000 GET       0004    000074 GRACTM    0003 R  000047 GREF      0000 I  000000 I         0006    000000 IBEGIN
    [003 I  000023 IDADJ     0003 I  000024 INDEX     0004 I  000066 INDXVL    0000    000051 INJP$     0006    000003 IPT
    [006    000130 IQ1       0000 I  000002 J         0003 I  000000 JUNK      0000 I  000001 KFIR      0006    000107 KLBC
    [000 I  000004 KLJ       0004    000051 KROOT     0006    000127 LBUF      0006    000121 LBUFD     0003 I  000020 LINK
    [004 I  000003 LKURT     0006    000102 LNCAT     0004 I  000007 LOSUM     0004 I  000004 LOVAR     0004 I  000002 LR
    [004 I  000006 LSKEW     0003 I  000021 LSUBS     0004 I  000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV
    [004    000002 LVRIN     0005 R  000012 MACCEL    0006    000122 MAXBF     0004 I  000001 MM        0004    000110 MODE
    [004 I  000000 MQ        0004 I  000011 MXAR      0003 I  000014 NARL      0006    000117 NBUFSD    0006    000120 NDUMP
    [004 I  000010 NINCLS    0006    000106 NIT       0006    000004 NOFLD     0006    000126 NPTS      0004 I  000070 NPTSO
    [003 I  000043 NSYMB     0003 I  000020 NTB       0003    000016 NTBSZM    0003 I  000015 NTOP      0003    000017 NWANT
    [006    000125 NWORS     0006    000124 NWOS      0003 R  000045 OCIN      0004 R  000055 ODCON     0003 R  000047 ODEN
    [003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW        0005 R  000010 PACCEL    0003 R  000050 PCOND
    [003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS     0003 R  000042 PORAT     0004    000071 PORATH
    [003 R  000026 PRIRCM    0006 I  000103 PRNT      0003 R  000031 PROP      0006 I  000111 PROUT     0006 I  000110 PRTME
    [003 R  000044 PST       0004    000065 SBLTH     0006 I  000114 SCRAM1    0004    000062 SEPTH     0005    000002 SKBND
    [005 R  000003 SKCHI     0004    000112 SPCOR     0003 R  000032 SPFAC     0004    000072 SPMVTH    0006 I  000005 SYM
    [006 I  000112 TOTPIX    0006 I  000001 TOTWRD    0005    000004 TRBND     0005 R  000005 TRCHI     0005    000006 URKBND
    [005 R  000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN     0004    000101 VOLLIM
    [003 R  000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R  000034 W         0003 R  000033 WADJ
```

```
C004 P 000060 WADJIN    0005   000000 WAIT     0004   000106 WDELSM    0004   000067 WFAC     0004   000105 WSIM
C004 R 00005C WTINIT    0007   000000 WTNEW    0004   000056 XOVFLO    0004   000057 XUNFLO

00101            SUBROUTINE CLINIT(KROT)                                                    000000
00101      C   THIS ROUTINE CONTAINS THE VARIOUS STATEMENTS NECESSARY TO                    000000
00101      C      INITIALIZE THE CLUSTERING ALGORITHM.                                      000000
00103            INCLUDE MISH                                                               000000
00131            INCLUDE CMBK10                                                             000000
00140            COMMON /INITL/WTNEW,DEVINI,CHANIN                                          000000
00141            CHIVAL(DF)=DF*(1.-.222/DF+CONLV*SQRT(.222/DF))**3                          000000
00142            DO 2  I=NTOP,NARL                                                       NEW000003
00145      2     LINK(I) = 0                                                             NEW000006
00147            AMQ=MQ                                                                     000007
00147      C  WE FIRST SET UP VARIOUS INDEX ARRAYS FOR A PARTICULAR                         000007
00147      C        NUMBER OF CHANNELS MQ.                                                  000007
00147      C       SET UP THE TRIANGULAR POSITION ARRAY MXAR.                               000007
00150            MM=0                                                                       000032
00151            DO 10 I=1,31                                                               000037
00154            MXAR(I)=MM                                                                 000037
00155         10 MM=MM+I                                                                    000041
00157            MM=MXAR(MQ+1)                                                              000046
00160            AMM=MM                                                                     000050
00160      C NOW WE SET UP THE ORIGIN VECTORS, LR AND LV, OF THE VARIOUS ARRAYS             000050
00160      C     AND VECTORS IN A CLUSTER NODE.                                             000050
00161            NINCLS=1                                                                   000070
00162            DO 21 I=1,NARRS                                                            000076
00165            LR(I)=NINCLS                                                               000076
00166         21 NINCLS=NINCLS+MM                                                           000077
00170            DO 22 I=1,NVECS                                                            000106
00173            LV(I)=NINCLS                                                               000106
00174         22 NINCLS=NINCLS+MQ                                                           000107
00176            NINCLS=NINCLS+NSCALS-1                                                     000113
00176      C  WE MUST ALSO SET UP SOME THRESHOLDS FOR USE BY THE STATISTICAL                000113
00176      C        SYSTEM.                                                                 000113
00177            SKCHI=(AMQ+2.)*(AMQ+4.)*CHIVAL(AMQ)                                        000115
00200            URKCHI=AMQ*(AMQ+4.)*(AMQ+6.)/(AMQ-.999)*CHIVAL(AMM-1.)                     000125
00201            TRCHI=CONLV*CONLV*(AMQ*(AMQ+2.)*(AMQ+3.)*8.)                               000143
00201      C  WE NOW CREATE THE HEAD NODE OF THE CLUSTER TREE.  THIS IS NOT                 000143
00201      C     AN ACTUAL CLUSTER, AND DOES NOT HAVE STORAGE FOR ANY                       000143
00201      C     OF THE STATISTICAL ARRAYS.                                                 000143
00202            NPTSO=0                                                                    000155
00203            KROT=GET(NSCALS)                                                           000156
00204            LINK(KROT)=-262139                                                         000163
00205            LSUPER(KROT)=-262142                                                       000165
00206            IDADJ(KROT)=999999                                                         000167
00207            INDEX(KROT)=-0                                                             000171
00210            SPFAC(KROT)=99999.                                                         000173
00211            W(KROT)=WTINIT                                                             000175
00212            OW(KROT)=W(KROT)                                                           000177
00213            PQRAT(KROT)=0.                                                             000200
00214            PROP(KROT)=1.                                                              000201
00215            OPROP(KROT)=1.                                                             000203
00216            CIN(KROT)=W(KROT)                                                          000205
00217            OCIN(KROT)=CIN(KROT)                                                       000207
```

```
00220      52*            CTOT(KROT)=0.                                                    000210
00221      53*            ODEN(KROT)=W(KROT)                                               000211
00222      54*            PRIRCM(KROT)=1.                                                  000213
00222      55*      C  NEXT THE INITIAL NODE IS SET UP, TOGETHER WITH SOME CONTROL THRESHOLDS.  000213
00223      56*         57 KFIR=GET(NINCLS)                                                 000215
00223      57*      C     PRINT 666,KFIR,NINCLS,NTBSZM                                     000215
00223      58*      C  666 FORMAT(' CLINIT:',5I6)                                          000215
00224      59*            DO 54 J=1,MM                                                     000221
00227      60*            GREF(KFIR+LOVAR+J)=0.                                            000235
00230      61*            GREF(KFIR+LKURT+J)=0.                                            000241
00231      62*         54 VRIN(KFIR+J)=0.                                                  000242
00233      63*            DEV2WT=DEVINI*DEVINI*WTINIT                                      000245
00236      64*            DO 53 J=1,MQ                                                     000251
00237      65*            GREF(KFIR+LSUM+J)=WTINIT*CHANIN                                  00026*
00240      66*            GREF(KFIR+LOSUM+J)=WTINIT*CHANIN                                 000272
00241      67*            KLJ=KFIR+MXAR(J+1)                                               000275
00242      68*            VRIN(KLJ)=1./DEV2WT                                              000301
00243      69*            GREF(KLJ+LOVAR)=DEV2WT                                           000305
00244      70*            GREF(KLJ+LKURT)=(MQ+2)*DEV2WT                                    000311
00245      71*         53 GREF(KFIR+LSKEW+J)=0.                                            000323
00247      72*            VOLRT(KFIR)=.9357622969E-13*(2.506628275*DEVINI)**MQ             000326
00250      73*            VOLIN(KFIR)=VOLRT(KFIR)**2                                       000337
00250      74*      C     VOLIN*EXP(DCON)=(2*PI)**MQ*DET(COVARIANCE)=(2*PI/W)**MQ/DET(WRIN)  000337
00251      75*            ODCON=MQ*ALOG(WTINIT)+60.                                        000341
00252      76*            DCON(KFIR)=ODCON                                                 000351
00253      77*            W(KFIR)=WTINIT                                                   000352
00254      78*            OW(KFIR)=WTINIT                                                  000354
00255      79*            CIN(KFIR)=W(KFIR)                                                000356
00256      80*            OCIN(KFIR)=CIN(KFIR)                                             000360
00257      81*            WADJ(KFIR)=WADJIN                                                000361
00260      82*            SPFAC(KFIR)=-9999.                                               000363
00261      83*            PORAT(KFIR)=0.                                                   000365
00262      84*            CTOT(KFIR)=0.                                                    000366
00263      85*            ODEN(KFIR)=W(KFIR)                                               000367
00264      86*            PROP(KFIR)=1.                                                    000371
00265      87*            OPROP(KFIR)=1.                                                   000373
00266      88*            PRIRCM(KFIR)=1.                                                  000375
00267      89*            LINK(KFIR)=0                                                     000377
00270      90*            LSJBS(KFIR)=0                                                    000400
00271      91*            LSUPER(KFIR)=KROT                                                000401
00272      92*            LSUBS(KROT)=KFIR                                                 000403
00273      93*            TOTPIX = TOTWRD/MQ                                           NEW000405
00274      94*            IDADJ(KFIR)=TOTPIX                                               000411
00275      95*            INDEX(KFIR)=INDXVL                                               000412
00276      96*            PRINT 273,MQ,CONLV,TRCHI,SKCHI,URKCHI,KROT,KFIR                  000414
00301      97*       273 FORMAT ('1 CONFIDENCE LEVELS',I4,' CHANNELS',F8.4,'  CHISQUARES',  000427
00301      98*           1    3E11.5/'    ROOT',I5,'    FIRST',I5)                        000427
00310      99*            RETURN                                                           000427
00311     100*            END                                                              000461

 END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,S CLPR,CLPR,CLPR
FOR SOE3-04/18/78-01:28:16 (0,0)

   SUBROUTINE CLPR       ENTRY POINT 000715

   STORAGE USED: CODE(1) 000736; DATA(0) 000364; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    GET
    0007    LOCK
    0010    SQMTX
    0011    MINV
    0012    FREE
    0013    NPRT$
    0014    NIO1$
    0015    NIO2$
    0016    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME).

    0000    000100 101F      0000    000172 102F      0000    000226 104F      0000    000253 105F      0000    000261 107F
    0000    000270 108F      0000    000320 109F      0000    000211 112F      0000    000234 113F      0000    000243 116F
    0001    000523 150L      0000    000300 163F      0000    000310 166F      0001    000657 200L      0001    000207 235G
    0001    000251 247G      0001    000325 262G      0001    000355 270G      0001    000365 276G      0001    000373 303G
    0001    000414 317G      0001    000422 325G      0001    000452 335G      0001    000460 343G      0001    000466 346G
    0001    000515 357G      0001    000565 370G      0001    000604 377G      0001    000633 405G      0001    000643 413G
    0001    000651 420G      0001    000167 5L        0003 R  000020 ALINK     0000 R  000036 AMEAN     0004    000064 AMM
    0004    000075 AMOFAC    0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ
    0004    000107 BETTER    0004    000102 BIAS      0003 R  000027 CIN       0005    000001 CONLV     0004    000111 CORLEN
    0003 R  000030 CTOT      0000 R  000073 CVL       0003 R  000041 DCON      0004    000053 DELT      0003 R  000043 DISS
    0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS       0003 R  000050 GEN       0006 I  000000 GET
    0004    000074 GRACTM    0003 R  000047 GREF      0000 I  000066 I         0003 I  000023 IDADJ     0003 I  000024 INDEX
    0004    000066 INDXVL    0000    000327 INJPS     0000 I  000074 J         0003 I  000000 JUNK      0004    000051 KROOT
    0000 R  000000 KTEMP     0000 I  000071 LA        0000 I  000072 LB        0000 I  000077 LCH       0003 I  000020 LINK
    0004    000003 LKURT     0007 I  000000 LOCK      0000 I  000067 LOD       0004 I  000007 LOSUM     0004 I  000004 LOVAR
    0003 I  000057 LPCC      0000 I  000060 LPCD      0000 I  000061 LPCDC     0004 I  000002 LR        0004    000006 LSKEW
    0003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV        0004    000002 LVRIN
    0005 R  000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I  000000 MQ        0000 I  000070 MQS
    0004 I  000011 MXAR      0000 I  000076 N         0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO
    0003 I  000043 NSYMB     0003 I  000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT
    0003 R  000045 ODIN      0004    000055 ODCON     0003 R  000047 ODEN      0000 R  000046 OMEAN     0003 R  000046 OPRIOR
    0003 R  000035 OPROP     0000 R  000065 OVSK      0003 R  000036 OW        0000 R  000063 OWUSE     0005 R  000010 PACCEL
    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS     0003 R  000042 PQRAT
    0004    000071 PQRATH    0003 R  000026 PRIRCM    0003 R  000031 PROP      0000 R  000075 PROUT     0000 R  000056 PRR
    0003 R  000044 PST       0004    000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI
    0004    000112 SPCOR     0003 R  000032 SPFAC     0004    000072 SPMVTH    0005    000004 TRBND     0005    000005 TRCHI
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C000  R 000064 TRK        0005    000006 URKBND     0005    000007 URKCHI     0005 R 000014 VACCEL     0004    000063 VFAC
C003  R 000037 VOLIN      0004    000101 VOLLIM     0003 R 000040 VOLRT     0003 R 000050 VRIN       0004    000104 VRJOIN
C003  R 000034 W          0003 R 000033 WADJ      0004    000060 WADJIN     0005    000000 WAIT      0004    000106 WDELSM
C004    000067 WFAC       0004    000105 WSIM       0004    000050 WTINIT     0000 R 000062 WUSE       0004    000056 XOVFLO
C004    000057 YUNFLO
```

```
00101     1*          SUBROUTINE CLPR(KL,IN,SUM,SKEW,KURT)                                                    000010
00101     2*      C  THIS ROUTINE PRINTS OUT ALL THE VARIABLES BELONGING TO SOME                              000010
00101     3*      C   CLASS INDEXED BY KL.                                                                    000010
00102     4*            INCLUDE MISH                                                                          000010
00131     5*            REAL KTEMP(30)                                                                        000010
00132     6*            REAL SUM(1),SKEW(1),KURT(1)                                                           000010
00133     7*            REAL AMEAN(MQMAX),QMEAN(MQMAX)                                                        000010
00136     8*            IF(KL.EQ.0) RETURN                                                                    000010
00136     9*            PRR=1.                                                                                000015
00137    10*            LPCC=LSUPER(KL)                                                                       000020
00140    11*            IF(INDEX(KL).NE.0) PRR=PROP(KL)/PRIRCM(LPCC)                                          000025
00142    12*            PRINT 101,IN,INDEX(KL),PRR,W(LPCC),SPFAC(KL),W(KL),OW(KL),                            000034
00142    13*           1 WADJ(KL),IDADJ(KL),PROP(KL),CIN(KL),CTOT(KL),OPROP(KL),OCIN(KL),                     000034
00142    14*           2   ODEN(KL),PORAT(KL),VOLIN(KL),VOLRT(KL),DCON(KL)                                    000034
00167    15*         101 FORMAT('CCLUSTER',I4,' INDEX',I4,'  PROPORTION',F9.6,                                000067
00167    16*           1    '  W*',F9.3,'  SPLIT',F9.4/                                                      000067
00167    17*           2  5X,'WEIGHT',F12.3,2X,'WAS',F12.3,                                                  000067
00167    18*           3     4X,'ADJUST',F12.3,'  ID',I6/                                                    000067
00167    19*           4   5X,'PROPORTION: PROP',F8.5,'  CIN',E12.5,'  CTOT',E12.5/5X,                       000067
00167    20*           5 'OLD PROP',F9.6,'  CIN',E10.5,'  ODEN',E10.5,'  DIFFER',E10.5/                       000067
00167    21*           6   5X,'VOLUME',E14.6,'  ROOT',E14.6,'  DCON',F13.7)                                  000067
00170    22*            LPCD=LINK(KL)                                                                         000067
00171    23*            LPCDC=LSUBS(KL)                                                                       000072
00172    24*            PRINT 102,KL,INDEX(LPCD),LINK(KL),INDEX(LPCDC),LSUBS(KL),                             000074
00172    25*           1   INDEX(LPCC),LSUPER(KL),NSYMB(KL)                                                   000074
00204    26*         102 FORMAT('    LOCATION',I5,'  LINK',I3,I5,'  SUBS',I3,I5,'  SUPER',                    000111
00204    27*           1  I3,I5,'  SYMBOL',I3)                                                               000111
00205    28*            PRINT 112,PST(KL),PCOND(KL),PCUM(KL),PRIRCM(KL)                                       000111
00213    29*         112 FORMAT(5X,'NET PROB',E10.5,'   DIRECT',E10.5,'   CUMS',                              000121
00213    30*           1    E10.5,' *',E10.5)                                                                000121
00214    31*            IF(PCUM(KL).LT.(ABS(10.**-25)).OR.PRIRCM(KL).LT.(ABS(10.**-25)))PR                    000121
00214    32*           * INT 104,PCUM(KL),PRIRCM(KL)                                                         000121
00221    33*         104  FORMAT(T46,'CUMS',I6,' * ',I6,//)                                                  000146
00222    34*            IF(INDEX(KL).EQ.0) RETURN                                                             000146
00226    35*            WUSE=W(KL)                                                                            000154
00225    36*            OWUSE=OW(KL)                                                                          000157
00226    37*            IF(INDEX(KL).GE.0) GO TO 5                                                            000161
00230    38*            WUSE=OW(KL)                                                                           000164
00231    39*            OWUSE=W(KL)                                                                           000165
00232    40*          5 TRK=0.                                                                                000167
00233    41*            OVSK=0.                                                                               000167
00234    42*            DO 2 I=1,MQ                                                                           000207
00237    43*            LOD=LOCK(I,I)                                                                         000207
00240    44*            TRK=TRK+KURT(KL+LOD)                                                                  000214
00241    45*            OVSK=OVSK+GREF(LOSUM+KL+I)**2                                                         000222
00242    46*            QMEAN(I)=GREF(LOSUM+I+KL)/OW(KL)                                                      000231
00243    47*          2 AMEAN(I)=SUM(I+KL)/W(KL)                                                              000234
00245    48*            PRINT 113,(AMEAN(I),I=1,MQ)                                                           000241
```

```
00251 1     49*      113   FORMAT ('0   MEAN',10X,8F13.6/(18X,8F13.6))                                  NEW000254
00251 1     50*      C                                                                                  -01000254
00251 4     51*            MQS=MQ*MQ                                                                       000254
00251 8     52*      C CHANGE RE:RASSBACH 3/21/77                                                          000254
00251 E     53*            LA=GET(MQS)                                                                     000257
00251 E     54*      C CHANGE RE:RASSBACH 3/21/77                                                          000257
00251 E     55*            LB=GET(MQS)                                                                     000263
00251 7     56*            CALL SQMTX(ALINK(LB),VRIN(KL+1))                                                000267
00261 0     57*            CALL MINV(ALINK(LA),ALINK(LB),ALINK(LB),CVL)                                    000300
00261 1     58*            DO 6 I=1,MQS                                                                    000326
00261 4     59*          6 ALINK(LA+I-1)=ALINK(LA+I-1)/WUSE                                                000326
00261 9     60*            PRINT 116,(ALINK(LA+J-1),J=1,MQ)                                                000331
00274 1     61*      116   FORMAT ('0   COVARIANCE      ',8F13.6/(18X,8F13.6))                          NEW000365
00275 5     62*            DO 7 I=2,MQ                                                                -01000365
00301 0     63*          7 PRINT 105, I,(ALINK(LA+MQ*I+J-MQ-1),J=1,MQ)                                     000365
00311 0     64*      105   FORMAT(11X,I5,2X,5E13.6/(16X,5E13.6))                                           000402
00311 0     65*      C                                                                                      000400
00311 0     66*      C                                                                                  NEW000400
00311 0     67*      C  SKIP PRINTING SKEW, KURT, OLD MEAN AND OLD COVARIANCE IS PROUT NOT 3            NEW000400
00311 1     68*            IF (PROUT .NE. 3) GO TO 200                                                 NEW000400
00311 1     69*      C                                                                                      000400
00311 3     70*            IF(TPK.EQ.0.) GO TO 150                                                         000403
00311 5     71*            PRINT 107,(SKEW(KL+I),I=1,MQ)                                                   000405
00322 1     72*      107   FORMAT ('0   SKEW(*W)',6X,8F13.6/(18X,8F13.6))                              NEW000422
00324 4     73*            DO 300 J=1,MQ                                                              -01000422
00327 1     74*            LOD=LOCK(1,J)                                                                   000422
00330 0     75*            N=LOD+KL                                                                        000427
00331 1     76*        300 KTEMP(J)=KURT(N)                                                                000433
00333 3     77*            PRINT 108, (KTEMP(J),J=1,MQ)                                                    000437
00341 1     78*      108   FORMAT ('0   KURT(*W)   I',2X,8F13.6/(16X,8F13.6))                          NEW000460
00342 2     79*            DO 8 I=2,MQ                                                                -01000460
00344 5     80*            DO 308 J=1,MQ                                                                   000466
00351 0     81*            LOD=LOCK(I,J)                                                                   000466
00351 1     82*            LCH=LOD+KL                                                                      000473
00352 3     83*        308 KTEMP(J)=KURT(LCH)                                                              000477
00354 4     84*          8 PRINT 105,I, (KTEMP(J),J=1,MQ)                                                  000503
00364 4     85*        150 IF(OVSK.EQ.0..OR.OPROP(KL).EQ.PROP(KL).AND.INDEX(KL).GT.0)                      000523
00364 4     86*       1     GO TO 200                                                                      000523
00366 6     87*            PRINT 163,(OMEAN(I),I=1,MQ)                                                     000544
00377 6     88*      163   FORMAT ('0     OLD MEAN',4X,8F13.6/(18X,8F13.6))                           NEW000570
00379 5     89*            CALL SQMTX(ALINK(LA),GREF(LOVAR+KL+1))                                     -01000570
00376 6     90*            DO 156 I=1,MQS                                                                  000604
00401 1     91*        156 ALINK(LA+I-1)=ALINK(LA+I-1)/OWUSE                                               000604
00403 3     92*            PRINT 166,(ALINK(LA+J-1),J=1,MQ)                                                000607
00411 1     93*      166   FORMAT('0   OLD COVARIANCE',1X,8F13.6/,(19X,8F13.6))                        NEW000643
00412 2     94*            DO 157 I=2,MQ                                                              -01000643
00415 5     95*        157 PRINT 105,I,(ALINK(LA+MQ*I+J-MQ-1),J=1,MQ)                                      000643
00425 5     96*        200 CALL FREE(LA,MQS)                                                               000657
00426 6     97*            CALL FREE(LB,MQS)                                                               000662
00427 7     98*            PRINT 109                                                                       000666
00431 1     99*        109 FORMAT(/)                                                                       000672
00432 2    100*            RETURN                                                                          000672
00433 3    101*            END                                                                             000735

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,S CLUST,CLUST,CLUST
FOR SOE3-04/18/78-01:28:21 (4,0)

   SUBROUTINE CLUST     ENTRY POINT 000314

   STORAGE USED: CODE(1) 000332; DATA(0) 000113; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016
    0006    BIGCOM  000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0007    ISPLIT
    0010    CORECT
    0011    DOTSQ
    0012    NWDUS
    0013    NI025
    0014    EXP
    0015    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000024 1000F      0001    000065 130L       0001    000127 131L       0001    000224 139L       0001    000030 147G
    0000    000036 2000F      0001    000170 531L       0000    000054 647F       0003 R  000020 ALINK      0004    000064 AMM
    0004    000075 AMOFAC     0004    000077 AMOMAX     0004    000076 AMOMIN     0004    000100 AMORAT     0004    000054 AMQ
    0004    000107 BETTER     0004    000102 BIAS       0006    000000 BIGDUM     0003 R  000027 CIN        0005    000001 CONLV
    0004    000111 CORLEN     0003 R  000030 CTOT       0003 R  000041 DCON       0004    000053 DELT       0003 R  000043 DISS
    0000 R  000021 DIST       0011 R  000000 DOTSQ      0004    000073 DWFAC      0004    000061 ELIMTH     0004    000052 EPS
    0003 R  000050 GEN        0004    000074 GRACTM     0003 R  000047 GREF       0003 I  000023 IDADJ      0000 I  000013 IDO
    0003 I  000024 INDEX      0004    000066 INDXVL     0000    000073 INJPS      0000 I  000015 ISEC       0007 L  000000 ISPLIT
    0000 I  000023 IW         0003 I  000000 JUNK       0000 I  000026 KFATH      0000 I  000016 KL         0000 I  000013 KMAX
    0004    000051 KROOT      0000 I  0E0010 KROT       0003 I  000020 LINK       0004    000003 LKURT      0004    000007 LOSUM
    0004    000004 LOVAR      0004 I  000002 LR         0004    000006 LSKEW      0003 I  000021 LSUBS      0004    000005 LSUM
    0003 I  000022 LSUPER     0004 I  000005 LV         0004    000002 LVRIN      0005 R  000012 MACCEL     0004    000001 MM
    0004    000110 MODE       0004 I  000000 MQ         0004 I  000011 MXAR       0003    000014 NARL       0000 I  000017 NCKPT
    0004    000010 NINCLS     0004    000070 NPTSO      0003 I  000043 NSYMB      0003 I  000020 NTB        0003    000016 NTBSZM
    0003    000015 NTOP       0003    000017 NWANT      0003 R  000045 OCIN       0004    000055 ODCON      0003 R  000047 ODEN
    0003 R  000046 OPRIOR     0003 R  000035 OPROP      0003 R  000036 OW         0000 R  000022 P          0005 R  000010 PACCEL
    0003 R  000050 PCOND      0003 R  000025 PCUM       0004    000103 PJOIN      0000 R  000012 PMAX       0003 R  000043 PPASS
    0003 P  000042 PQRAT      0004    000071 PQRATH     0003 R  000026 PRIRCM     0003 R  000031 PROP       0003 R  000044 PST
    0000 R  000014 PTCT       0000 R  000000 REL        0004    000065 SBLTH      0004    000062 SEPTH      0005    000002 SKBND
    0005    000003 SKCHI      0004    000112 SPCOR      0003 R  000032 SPFAC      0004    000072 SPMVTH     0005    000004 TRBND
    0005    000005 TRCHI      0005    000006 URKBND     0005    000007 URKCHI     0005 R  000014 VACCEL     0004    000063 VFAC
    0003 R  000037 VOLIN      0004    000101 VOLLIM     0003 R  000040 VOLRT      0003 R  000050 WRIA       0004    000104 VRJOIN
    0003 R  000034 W          0003 R  000033 WADJ       0004    000060 WADJIN     0005    000000 WAIT       0004    000106 WGELSM
    0004    000067 WFAC       0004    000105 WSIM       0004    000050 WTINIT     0004    000056 XOVFLO     0004    000057 XUNFLO
```

```
00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            SUBROUTINE CLUST(BIGP,NDO,KLOUT,KROTIN,SUM)                                    000002
00101      2*      C    THIS PROGRAM TAKES EACH INPUT POINT AND CLASSIFIES IT,                          000002
00101      3*      C       FOR THE PURPOSE OF GENERATING A MAP.                                         000002
00101      4*      C                                                                                    000002
00101      5*      C  CLUST ARGUMENTS (CLUST DRAWN FROM STATIS)                                         000002
00101      6*      C     BIGP      INPUT DATA VECTOR                                                    000002
00101      7*      C     NDO       NO. DATA POINTS                                                      000002
00101      8*      C     KLOUT     KL OF OUTPUT CLASS                                                   000002
00101      9*      C     KROTIN    ROOT VERTEX                                                          000002
00101     10*      C     SUM       POSITION OF SUM VECTOR IN CLUSTER.                                   000002
00101     11*      C           OUTPUT SYMBOL IS DERIVED FROM NSYMB(KL)                                  000002
00101     12*      C                                                                                    000002
00103     13*            INCLUDE MISH                                                                   000002
00131     14*            DIMENSION BIGP(MQ,NDO),KLOUT(NDO),SUM(1)                                       000002
00132     15*            COMMON/BIGCOM/ BIGDUM                                                          000002
00133     16*            REAL REL(MQMAX)                                                                000002
00134     17*            LOGICAL ISPLIT                                                                 000002
00135     18*            XP(DIST)=EXP(-.5*DIST)                                                         000002
00136     19*            KROT=KROTIN                                                                    000002
00136     20*      C                                                                                    000002
00137     21*            IF ( KROT .LE. 0 )  WRITE(6,1000)  KROT                                        000004
00143     22*       1000 FORMAT(     2X,'::::::: WARNING :::::::  IN CLUST,  KROT=', 3X,                000015
00143     23*           *  I10      )                                                                   000015
00146     24*            IF (KROT .LE. 0) RETURN                                                        000015
00146     25*      C                                                                                    000015
00146     26*            DO 399 IDO=1,NDO                                                               000023
00146     27*      C  INSPECT EACH POINT                                                                000023
00151     28*            PMAX=-1.                                                                       000030
00152     29*            KMAX=0                                                                         000032
00152     30*      C  CHANGE RE:RASSBACH 3/21/77                                                        000032
00152     31*      C  USES PCOND FOR PPASS, DIST FOR DISS(KL)                                           000032
00153     32*            PCOND(KROT)=1.                                                                 000033
00154     33*            PTOT=0.                                                                        000034
00155     34*            ISEC=0                                                                         000035
00156     35*            KL=LSUBS(KROT)                                                                 000036
00156     36*      C                                                                                    000036
00157     37*            NCKPT = 1                                                                      000040
00157     38*      C                                                                                    000040
00160     39*            IF ( KL .LE. 0 )  WRITE(6,2000)   NCKPT , KL                                   000042
00165     40*       2000 FORMAT(     2X,'::::::::::: WARNING ::::::::  IN CLUST, AT CHECKPOIN          000054
00165     41*           *T' , 2X, I3, 3X, ', KL=' , I8      )                                           000054
00166     42*            IF (KL .LE. 0) RETURN                                                          000054
00166     43*      C                                                                                    000054
00170     44*            KPATH=KROT                                                                     000062
00170     45*      C   GO DOWN CLUSTER TREE                                                             000062
00170     46*      C CHANGE RE:RASSBACH 3/21/77                                                         000062
00171     47*        130 PCOND(KL)=PROP(KL)/(PRIROM(KPATH))*PCOND(KPATH)                                000065
00172     48*            IF(.NOT.ISPLIT(KL)) GO TO 131                                                  000072
00174     49*            KPATH=KL                                                                       000076
00175     50*            KL=LSUBS(KL)                                                                   000101
00176     51*            NCKPT = 2                                                                      000103
00176     52*      C                                                                                    000103
00177     53*            IF ( KL .LE. 0 )  WRITE(6,2000)  NCKPT , KL                                    000105
00204     54*            IF (KL .LE. 0) RETURN                                                          000117
00206     55*            GO TO 130                                                                      000125
```

```
00207      56*          131 CALL CORECT(REL,BIGP(1,IDO),W(KL),SUM(KL+1))                    000127
00207      57*        C  CHANGE RE:RASSBACH 3/21/77                                         000127
00210      58*             DIST=DOTSQ(REL,VRIN(KL+1))*W(KL)                                 000151
00211      59*             IF(ABS(DIST+DCON(KL)).LE. 160.) GO TO 531                        000161
00213      60*             GO TO 139                                                        000166
00214      61*          531 CONTINUE                                                        000171
00214      62*        C  CHANGE RE:RASSBACH 3/21/77                                         000171
00215      63*             P=EXP(DIST+DCON(KL))/VOLRT(KL)*PCOND(KL)                         000179
00216      64*             PTOT=PTOT+P                                                      000202
00217      65*             IF(P.LE.PMAX.OR.ISPLIT(KL))                                      000204
00217      66*            1   GO TO 139                                                     000204
00221      67*             PMAX=P                                                           000217
00222      68*             KMAX=KL                                                          000221
00223      69*          139 KL=LINK(KL)                                                     000224
00224      70*             IF(KL)130,149,130                                                000226
00224      71*        C    GO UP TREE                                                      000226
00224      72*        C  CHANGE RE:RASSBACH 3/21/77                                         000226
00225      73*        C 149 PCOND(KL) = 0                                                   000226
00225      74*        C                                                                     000226
00227      75*          149  KL = KPATH                                                    000227
00227      76*        C                                                                     000227
00230      77*             PCOND(KL) = 0                                                    000232
00231      78*             KPATH=LSUPER(KL)                                                 000233
00232      79*             NCKPT = 3                                                        000235
00232      80*        C                                                                     000235
00233      81*             IF ( KL .LE. 0 )  WRITE(6,2000)  NCKPT, KL                       000237
00240      82*             IF (KL .LE. 0) RETURN                                            000251
00240      83*        C                                                                     000251
00242      84*             IF(KL.NE.KROT) GO TO 131                                         000257
00244      85*          309 IW=KLOUT(IDO)                                                   000262
00245      86*             KLOUT(IDO)=KMAX                                                  000264
00246      87*             IF(PTOT.NE.0.) PMAX=PMAX/PTOT                                    000266
00246      88*        C     D        IF(II.LT.ITLIM) GO TO 399                            000266
00246      89*        C     D  646 PRINT 647,IDO,W(KROT),KL,ISEC,(KTR(I),I=1,ITLIM)       000266
00250      90*          647 FORMAT('CLOOP IN CLUST:IDO,W(KROT),KL,SECTION',I5,E11.5,2I5   000276
00250      91*            1    /(1X14I5))                                                 000276
00251      92*          399 CONTINUE                                                       000276
00252      93*             RETURN                                                          000276
00254      94*             END                                                             000331

      END OF COMPILATION:      1 DIAGNOSTICS.
```

```
@FOR,US  CLUSMP,CLUSMP,CLUSMP
FOR SOE3:-04/18/78-01:28:26 (4,5)

   SUBROUTINE CLUSMP    ENTRY POINT 001227

   STORAGE USED: CODE(1) 001357; DATA(0) 000244; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    ARRAY   037201
    C004    INFORM  000407
    C005    CLUSTR  000131
    C006    CLUS    002017
    C007    MISC    000113
    C010    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C011    FDLINT
    C012    CLUST
    C013    RREAD
    C014    NWDU$
    C015    NIO2$
    C016    NIO1$
    C017    NERR2$
    C020    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000162 100L     0001    000164 110L     0001    001314 1110L    0001    001327 1115L    0001    001340 1120L
    C001    000356 140L     0001    000361 150L     0001    000416 152L     0001    000503 163L     0001    000007 164G
    C001    000521 165L     0001    000533 167L     0001    000535 170L     0001    000625 175L     0001    000027 177G
    C001    000713 177L     0001    000727 180L     0001    000760 183L     0001    000765 184L     0001    000771 185L
    C001    001002 190L     0001    001010 200L     0001    000045 206G     0001    000060 214G     0001    000065 221G
    C001    000122 236G     0001    000227 264G     0001    000234 270G     0000    000102 275F     0001    000654 285L
    C001    000253 303G     0001    000276 307G     0001    001044 400L     0001    000562 407G     0001    000643 432G
    C001    000651 440G     0001    000073 50L      0000    000065 500F     0000    000066 510F     0000    000100 520F
    C001    001056 526G     0001    001100 537G     0000    000105 570F     0001    001171 576G     0000    000124 590F
    C000    000130 630F     0000    000140 750F     0000    000147 760F     0000    000155 770F     0000    000165 780F
    C001    000113 80L      0000 I  000024 ADFLD    0000 I  000011 ADRESS   0006 R  000020 ALINK    0007 I  000064 AMM
    C007 I  000075 AMOFAC   0007 I  000077 AMOMAX   0007 I  000076 AMOMIN   0007 I  000100 AMORAT   0007 I  000054 AMO
    C005 I  000123 AREA     0003 I  000000 ARRAY    0004 I  000351 BESTVC   0007 I  000107 BETTER    0007 I  000102 BIAS
    C000 I  000021 BLANK    0000 I  000054 BN       0000 I  000062 BNS      0000 I  000063 BUF      0000 I  000051 BUFAD
    C000 I  000017 BUFCT    0003 R  005670 BUFER    0005 I  000115 BUFPIX   0005 I  000116 BUFTOT   0006 R  000027 CIN
    C005 I  000002 CLSNAM   0003 I  004715 CLUSTN   0003 I  003720 COL      0010 I  000001 CONLV    0007 I  000111 CORLEN
    C006 R  000030 CTOT     0004 I  000053 DATAPE   0006 R  000041 DCON     0007 I  000053 DELT     0006 R  000043 DISS
    C004 I  000112 DIVSIZ   0005 I  000113 DUMP     0004 I  000062 DUPSYM   0007 I  000073 DWFAC    0007 I  000061 ELIMTH
    C000 I  000035 ENCLM    0007 I  000052 EPS      0004 I  000217 FETVC2   0004 I  000161 FETVEC    0004 I  000071 FILESV
    C000 I  000001 FL       0000 I  000171 FLDINF   0006 R  000050 GEN      0007 I  000074 GRACTM   0006 R  000047 GREF
    C004 I  000000 HEAD     0004 I  000255 HISVEC   0000 I  000022 I        0000 I  000044 IB       0005 I  000000 IBEGIN
    C004 I  000123 IBLOCK   0000 I  000050 IBO      0000 I  000056 IBS      0006 I  000023 IDADJ    0000 I  000045 IE
    C000 I  000055 IES      0000 I  000030 IFLD     0006 I  000024 INDEX    0007 I  000066 INDXVL   0000    000211 INJP$
    C000    000205 INJP$    0004 I  000313 INVERT   0005 I  000003 IPT      0005 I  000130 IQ1      0000 I  000013 ISTAT
```

```
0000 I 000036 J          0000 I 000047 JMP        0006 I 000090 JUNK       0000 I 000060 K          0004 I 000113 KEEPLV
0005 I 000107 KLBC       0007 I 000051 KROOT      0000 I 000057 L          0005 I 000127 LBUF       0005 I 000121 LBUFD
0000 I 000040 LINE       0000 I 000172 LINEND     0000 I 000173 LINEINC    0006 I 000020 LINK       0000 I 000171 LINSTR
0007   000003 LKURT      0005 I 000102 LNCAT      0007   000007 LOSUM      0007   000004 LOVAR      0000 I 000037 LPTS
0007 I 000002 LR         0007   000006 LSKEW      0006 I 000021 LSUBS      0007 I 000005 LSUM       0006 I 000022 LSUPER
0007 I 000005 LV         0007   000002 LVRIN      0010 R 000012 MACCEL     0004 I 000052 MAPTAP     0005 I 000122 MAXBF
0004 I 000072 MAXCLS     0004 I 000064 MAXDIV     0004 I 000055 MAXFET     0004 I 000074 MAXFLD     0004 I 000065 MINDIV
0007 I 000001 MM         0007 I 000110 MODE       0007 I 000000 MQ         0007 I 000011 MXAR       0000 I 000026 MXBF
0006 I 000014 NARL       0003 I 005073 NBLK       0003 I 005251 NBLKT      0005 I 000117 NBUFSD     0005 I 000120 NDUMP
0000 L 000000 NFIN       0000 I 000042 NI         0007 I 000010 NINCLS     0005   000106 NIT        0004 I 000073 NOCLS2
0004 I 000121 NOCLSS     0003 I 000043 NOEX       0004 I 000100 NOFEAT     0004 I 000101 NOFET2     0004 I 000102 NOFET4
0005 I 000004 NOFLD      0004 I 000075 NOFLD2     0004 I 000076 NOFLD3     0004 I 000111 NOGRP      0004 I 000110 NOHIST
0004 I 000107 NOSPEC     0004 I 000077 NOTRFD     0000 I 000015 NOWRD      0003 I 000053 NPNTS      0005 I 000126 NPTS
0007 I 000070 NPTS0      0000 I 000052 NSETS      0006 I 000043 NSYMB      0006 I 000020 NTB        0006 I 000016 NTBSZM
0006 I 000015 NTOP       0000 I 000061 NUM        0000 I 000032 NV         0006 I 000017 NWANT      0005 I 000125 NWORS
0005 I 000124 NWDS       0006 R 000045 OCIN       0007 I 000055 ODCON      0006 R 000047 ODEN       0006 R 000046 OPRIOR
0006 R 000035 OPROP      0003 I 004537 OUT        0006 R 000036 OW         0010 R 000010 PACCEL     0004 I 000056 PAGSIZ
0006 R 000050 PCOND      0004 I 000122 PCTSZ      0006 R 000025 PCUM       0007 I 000103 PJOIN      0006 R 000043 PPASS
0006 R 000042 PQRAT      0007 I 000071 PQRATH     0006 R 000026 PRIRCM     0005 I 000103 PRNT       0006 R 000031 PROP
0005 I 000111 PROUT      0004 I 000114 PRTLEV     0005 I 000110 PRTME      0006 R 000044 PST        0000 I 000175 SAMEND
0000 I 000176 SAMINC     0000 I 000041 SAMPS      0000 I 000174 SAMSTR     0004 I 000054 SAVTAP     0007 I 000065 SBLTH
0005 I 000114 SCRAMI     0007 I 000062 SEPTH      0004 I 000067 SERIAL     0010 I 000002 SKBND      0010 I 000003 SKCHI
0004 I 000120 SPCBAS     0007 I 000112 SPCOR      0006 R 000032 SPFAC      0004 I 000066 SPLMAX     0007 I 000072 SPMVTH
0000 I 000034 STCLM      0005 I 000005 SYM        0004 I 000057 TAPCHK     0004 I 000070 TAPESV     0004 I 000063 THPSYM
0000 I 000046 TNPTS      0003 I 037200 TOP        0005 I 000112 TOTPIX     0000 I 000064 TOTPTS     0000 I 000033 TOTSAM
0005 I 000001 TOTWRD     0010 I 000004 TRBND      0010 I 000005 TRCHI      0004 I 000060 TRNSYM     0004 I 000061 TSISYM
0000 I 000027 TWRD       0010 I 000006 URKBND     0010 I 000007 URKCHI     0010 R 000014 VACCEL     0004 I 000103 VARSIZ
0004 I 000104 VARSZ2     0004 I 000105 VARSZ4     0007 I 000063 VFAC       0006 R 000037 VOLIN      0007 I 000101 VOLLIM
0006 R 000040 VOLRT      0006 R 000050 VRIN       0007 I 000104 VRJOIN     0006 R 000034 W          0006 R 000033 WADJ
0007 I 000060 WADJIN     0010 I 000000 WAIT       0007 I 000106 WDELSM     0007 I 000067 WFAC       0000 I 000025 WRDCT
0000 I 000023 WRDSV      0007 I 000105 WSIM       0007 I 000050 WTINIT     0004 I 000116 XHGH       0004 I 000117 XLOW
0007 I 000056 XOVFLO     0004 I 000106 XSIZ       0000 I 000031 XTRA       0007 I 000057 XUNFLO     0004 I 000115 YSIZ
```

```
0000C  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101        1*            SUBROUTINE CLUSMP                                                    NEW000000
00101        2*       C*                                                                        -01000000
00101        3*       C*    THE PURPOSE OF CLUSMP IS TO PRINT THE CLUSTER MAP.  THE CLUSTER       000000
00101        4*       C*    MAP HAS EACH PIXEL REPRESENTED BY A SYMBOL.  EACH SYMBOL              000000
00101        5*       C*    REPRESENTS A CLUSTER TYPE                                             000000
00101        6*       C*                                                                          000000
00101        7*            IMPLICIT INTEGER (A-Z)                                                000000
00101        8*       C                                                                         NEW000000
00104        9*            COMMON /ARRAY/ ARRAY(16000),TOP                                      NEW000000
00104       10*       C                                                                         NEW000000
00104       11*       C                                                                         -01000000
00105       12*            INCLUDE COMBK1                                                        000000
00107       13*            INCLUDE CMBK10                                                        000000
00111       14*            INCLUDE MISH                                                          000000
00111       15*       C                                                                          000000
00114       16*            LOGICAL NFIN                                                          000000
00115       17*            REAL BUFER                                                            000000
00114       18*            DIMENSION BUFER(3300,2),COL(3,110),OUT(110),FL(8),FLOINF(6),          000000
00114       19*           *  CLUSTR(110),ADRESS(2),ISTAT(2),NBLK(MAXPOP),NBLKT(MAXPOP),          000000
00114       20*           *  NOWRD(2),BUFCT(2)                                                   000000
```

```
C
      EQUIVALENCE (COL(1,1),ARRAY(2001))
      EQUIVALENCE (OUT(1)   ,ARRAY(2400))
      EQUIVALENCE (CLUSTN(1),ARRAY(2510))
      EQUIVALENCE (NBLK(1),ARRAY(2620))
      EQUIVALENCE (NBLKT(1),ARRAY(2730))
      EQUIVALENCE (BUFER(1,1),ARRAY(3001))
      EQUIVALENCE (FLDINF(1),LINSTR) ,(FLDINF(4),SAMSTR),
     1            (FLDINF(2),LINEND) ,(FLDINF(5),SAMEND),
     2            (FLDINF(3),LININC) ,(FLDINF(6),SAMINC)
C*    FIELD INFORMATION STORED AS FOLLOWS
C*
C*    ARRAY(1)    =FIRST FIELD NAME FOR THIS CLASS
C*          (2)   =NO. OF VERTICES FOR THIS FIELD (NV)
C*          (3)-(3+NV*2)  = ACTUAL  VERTEX NUMBERS
C*          (3+NV*2)  =TOTAL PIXELS FOR THIS FIELD
C*          (4+NV*2)-(10+NV*2) = FLDINF BLOCK FOR THIS FIELD
C*
      DATA BLANK/' '/
C
C
C                    *****  INITIALIZE *****
      LNCAT=0
      ADRESS(1)=IBEGIN
      IPT=1
      DO 25 I=1,MAXPOP
   25 NBLKT(I)=0
C  WRDSV =
      WRDSV=0
C  ADFLD = 1ST DISK ADDRESS
      ADFLD=IBEGIN
C  WRDCT = TOTAL WORDS READ
      WRDCT=0
C  MXBF = MA  BUFFER SIZE
      MXBF=3300
C  TWRD = TOTAL WORDS LEFT TO BE READ
      TWRD=TOTWRD
C*
C                    ***** READ 2 BUFFERS OF DATA *****
      CALL BFINIT
C
C
C
C                    **********
      DO 600 IFLD=1,NOFLD
C
C
C  XTRA = SEGMENT  LREADY PROCESSED
      XTRA=0
C  NFIN = FALSE IF ONLY 1 PAGE NEEDED
      NFIN=.FALSE.
C  NV = NO OF VERTICES FOR THIS FIELD
      NV=ARRAY(IPT+1)
C  TOTSAM = TOTAL PIXELS FOR THIS FIELD
      TOTSAM=ARRAY(IPT+2+NV*2)
C
C  MOVE DATA DEFINING LINES AND SAMPLES
```

```
      DO 30 I=1,6
      FLDINF(I)=ARRAY(IPT+2+I+NV*2)
   30 CONTINUE
C
C  BLANK OUTPUT BUFFER
      DO 40 I=1,110
   40 OUT(I)=BLANK
C
C  ZERO COUNT OF POINTS IN CLUSTER
      DO 45 I=1,MAXPOP
   45 NBLK(I)=0
C*
C*    CHECK IF ALL OF CLUSTER MAP CAN FIT ACROSS ONE PAGE--ONLY 110
C*    SYMBOLS ARE PRINTED ACROSS THE PAGE FOR EVERY LINE.  THE PROGRAM
C*    WILL PRINT THE ENTIRE CLUSTER MAP IN 110 SYMBOL SEGMENTS
C*
C
C  SET STARTING ADDRESS AND ENDING ADDRESS FOR LINE
      STCLM=SAMSTR
      ENCLM=SAMEND
C
C  CK FOR MORE THAN 110 SEGMENTS SPECIFIED AND RESET MAXIMUM IF NECESSRY
C  NFIN = FALSE, IF 1 LINE    TRUE, IF 2 OR MORE LINES
C
   50 IF(((ENCLM-SAMSTR)/SAMINC+1-XTRA).LE. 110) GO TO 80
      ENCLM=  (109+XTRA)*SAMINC + SAMSTR
      NFIN=.TRUE.
C
C                    *** SET COLUMN HEADINGS ***
   80 CONTINUE
      J=0
      DO 100 I=SAMSTR,SAMEND,SAMINC
      IF( I .LT. STCLM)GO TO 100
      IF( I .GT. ENCLM)GO TO 110
      J=J+1
      COL(1,J)=I/100
      COL(2,J)=MOD(I,100)/10
      COL(3,J)=MOD(I,10)
  100 CONTINUE
C
C                    *** WRITE HEADINGS ***
  110 LPTS=J
      WRITE(6,500)
      WRITE(6,HEAD)
      WRITE(6,510)ARRAY(IPT),TOTSAM
C*.   PRINT COLUMN NUMBERS FOR CLUSTER MAP
      DO 120 I=1,3
  120 WRITE(6,520)(COL(I,J),J=1,LPTS)
      WRITE(6,500)
  500 FORMAT(/)
  510 FORMAT(//2X,A6,//' TOTAL NUMBER OF POINTS IN THIS FIELD',I7)
  520 FORMAT(9X,110I1)
C
C
C                ******  PROCESS ONE LINE OF DATA  ******
C
      DO 300 LINE=LINSTR,LINEND,LININC
```

```
00302     135*      C*
00302     136*      C*    CALL FDLINT TO OBTAIN FIELD INTERSECTIONS FOR THIS LINE
00305     137*            CALL FDLINT(ARRAY(IPT+2),NV,FL,LINE,SAMPS,NI)
00305     138*      C
00305     139*      C                    *****  PROCESS EACH INTERCEPT *****
00306     140*            DO 200 I=1,NI,2
00311     141*            NOEX=0
00311     142*      C
00311     143*      C*    SAVE THE BEGINNING AND END NUMBERS OF THIS INTERCEPT FOR ARRAY OUT
00311     144*      C*    WHICH IS PRINTED
00312     145*            IB =(FL(I)-SAMSTR)/SAMINC +1
00312     146*            IE =(FL(I+1)-SAMSTR)/SAMINC+1
00314     147*            IF(MOD(SAMSTR,SAMINC) .NE. MOD(FL(I),SAMINC))IB =IB +1
00316     148*            TNPTS=(IE-IB+1)*NOFEAT
00317     149*            IF(IB  .GT. IE )TNPTS=0
00321     150*            IF(IB  .GT. IE ) GO TO 140
00321     151*      C
00321     152*      C*    CHECK IF INTERCEPTS ARE WITHIN PRINTOUT LIMITS
00323     153*            IF(FL(I) .GT. ENCLM) GO TO 140
00325     154*            IF(FL(I+1) .LT. STCLM) GO TO 140
00327     155*            GO TO 150
00327     156*      C
00327     157*      C*    THESE CARDS ARE USED TO SET UP THE OUTPUT FOR BLANK LINES OR BLANK
00327     158*      C*    SPACES OR AREAS OUTSIDE OF PRINT LIMITS
00330     159*        140 CONTINUE
00331     160*            JMP=1
00332     161*            GO TO 175
00332     162*      C
00332     163*      C
00333     164*        150 CONTINUE
00333     165*      C*    RESAVE BEGINNING AND END NUMBERS FOR ARRAY OUT IF INTERCEPT(S)
00333     166*      C*    EXCEEDS PRIN  LIMIT
00334     167*            IF(FL(I) .GE. STCLM) GO TO 152
00336     168*            IBO=IB
00337     169*            IB=(STCLM-SAMSTR)/SAMINC+1
00340     170*            IF(MOD(SAMSTR,SAMINC) .NE. MOD(STCLM,SAMINC))IB=IB+1
00340     171*      C
00340     172*      C*    STORE NUMBER OF EXTRA POINTS THAT ARE IN INTERCEPT BUT ARE
00340     173*      C*    OUTSIDE THE PRINT LIMITS ON LEFT SIDE
00342     174*            NOEX=(IB-IBO)*NOFEAT
00343     175*            BUFAD=BUFAD+NOEX
00344     176*        152 IF(FL(I+1) .GT. ENCLM)IE=(ENCLM-SAMSTR)/SAMINC+1
00348     177*      C
00344     178*      C*    SET PRINT LIMITS IN THE 1-110 LIMITS WHEN THE NUMBERS WOULD EXCEED
00348     179*      C*    110 ON ANOTHER PASS THROUGH THE DATA
00346     180*            IB=IB-XTRA
00347     181*            IE=IE-XTRA
00350     182*            IF(IB .GT. IE) GO TO 140
00352     183*            NSETS=IE-IB+1
00353     184*            NPNTS=NSETS*NOFEAT
00352     185*      C
00354     186*        155 CONTINUE
00354     187*      C*
00354     188*      C*    CHECK IF NEEDED DATA IN THIS INTERCEPT IS IN TWO BUFFERS
00354     189*      C*
00355     190*            JMP=1
00356     191*            IF(BUFCT(BN)+NOEX+NPNTS .LE. NOWRD(BN))GO TO 170
```

```
00035E   192*      C
00035E   193*      C  NO. PTS PROCESSED + NO PTS TO LEFT + NO. PTS IN INTERCEPT   GE
00035E   194*      C     BUFFER SIZE, THEREFORE SOME PTS IN INTERCEPT ARE TO THE LEFT
00035E   195*      C*      SAVE THE DATA LIMITS FOR THE NEXT BUFER AND COMPUTE THE NUMBER OF
00035E   196*      C*      SETS OF POINTS IN THIS BUFFER
00036C   197*            NPNTS=NOWRD(BN)-(BUFCT(BN)+NOEX)
00036J   198*            JMP=3
00036Z   199*            IF(NPNTS)160,285,163
00036Z   200*      C
00036Z   201*      C  FIRST INTERCEPT PT IN THIS LINE DOES NOT OCCUR IN THIS BUFFER, SKIP
00036Z   202*      C     OVER POINTS IN THIS BUFFER
00036Z   203*      C*      THE EXTRA POINTS ARE IN BOTH BUFFERS
00036S   204*        160 JMP=4
00036E   205*            NOEX=(BUFCT(BN)+NOEX)-NOWRD(BN)
00036J   206*            NPNTS=NSETS*NOFEAT
00037C   207*            GO TO 285
00037C   208*      C
00037C   209*      C  FIRST INTERCEPT PT IN THIS LINE OCCURS IN THIS BUFFER
00037J   210*        163 NSETS=NPNTS/NOFEAT
00037J   211*      C*      SAVE THE NUMBER OF POINTS THAT ARE IN THE NEXT BUFFER IF DATA IS
00037J   212*      C*      SPLIT IN TWO BUFFERS
00037Z   213*            IES=IE
00037J   214*            IE=IB+NSETS-1
00037A   215*            IBS=IE+1
00037S   216*            JMP=2
00037E   217*            GO TO 170
00037E   218*      C
00037E   219*      C*      CALCULATE POINTS IN SECOND BUFER
00037J   220*        165 NSETS=IES-IBS+1
00040C   221*            NPNTS=NSETS*NOFEAT
00040J   222*            IE=IES
00040Z   223*            IB=IBS
00040J   224*        167 JMP=1
00040J   225*      C
00040J   226*      C*      CALL CLUST TO OBTAIN THE CLUSTER SUBSCRIPT SO THAT THE CLUSTER
00040J   227*      C*      SYMBOLS CAN BE COMPUTED FOR EACH SET OF FL'S WITHIN THE
00040J   228*      C*      START(STCLM) AND END(ENCLM)
00040A   229*        170 CALL CLUST(BUFER(BUFAD,BN),NSETS,CLUSTN,KLBC,GEN(LSUM))
00040A   230*      C
00040S   231*            L=0
00040S   232*      C*
00040S   233*      C*      STORE SYMBOLS FOR OUTPUT
00040A   234*            DO 173 K=IB,IE
00041J   235*            L=L+1
00041Z   236*            NUM=CLUSTN(L)
00041Z   237*      C     SET SYMBOL--THE SUBSCRIPT FOR SYM IS RESET TO 1 THROUGH MAXPOP
00041J   238*            J=MOD(NSYMB(NUM)-1,MAXPOP)+1
00041A   239*            IF ( J .LE. 0 ) J = 47
00041E   240*            LNCAT=MAXO(LNCAT,J)
00041J   241*            OUT(K)=SYM(J)
00041J   242*      C*      SAVE THE NUMBER OF PIXELS ASSIGNED TO THIS CLUSTER
00042C   243*        173 NBLK(J)=NBLK(J)+1
00042C   244*      C
00042Z   245*            IF(JMP .NE. 1) GO TO 285
00042Z   246*      C
00042A   247*        175 CONTINUE
00042S   248*            IF(I+1 .NE. NI) GO TO 285
```

```
                249*      C
                250*      C                  *** PRINT LINE OF OUTPUT AND BLANK BUFFER ***
                251*            WRITE(6,275)LINE ,(OUT(K),K=1,LPTS)
                252*        275 FORMAT(2X,I5,2X,110A1)
                253*            DO 280 K=1,110
                254*        280 OUT(K)=BLANK
                255*      C
                256*        285 CONTINUE
                257*            IF(BUFCT(BN)+TNPTS .LT. NOWRD(BN))GO TO 185
                258*      C
                259*      C           ** NO MORE DATA IN THIS BUFFER, PREVIOUS PTS + TOTAL PTS
                260*      C              IN INTERCEPT EXCEED END OF BUFFER.  SWITCH BUFFERS AND
                261*      C              START LOOK AHEAD READ IF THERE IS MORE DATA  **
                262*            TNPTS=TNPTS-(NOWRD(BN)-BUFCT(BN))
                263*            BUFCT(BN)=NOWRD(BN)
                264*            BNS=BN
                265*      C
                266*      C*          SET BUFFER NUMBER
                267*            BN=BN+1
                268*            IF(BN .GT. 2)BN=1
                269*            BUFAD=1
                270*            BUFCT(BN)=0
                271*            IF(JMP .NE. 4)GO TO 177
                272*            BUFAD=NDEX+1
                273*      C
                274*      C*          CHECK IF ALL DATA HAS BEEN READ
                275*        177 IF(WRDCT .EQ. TOTWRD)GO TO 183
                276*            IF(WRDCT +NOWRD(BNS) .LE. TOTWRD)GO TO 180
                277*            NOWRD(BNS)=TOTWRD-WRDCT
                278*      C
                279*      C  UPDATE TOTAL NO OF WORDS TO READ AND DISK ADRES FOR LOOK AHEAD READ.
                280*        180 WRDCT=WRDCT + NOWRD(BNS)
                281*            ADRESS(BNS)=ADRESS(BN)+ NOWRD(BN)
                282*      C
                283*            CALL ?READ(ADRESS(BNS),BUFER(1,BNS),NOWRD(BNS),ISTAT(BNS))
                284*            GO TO 184
                285*      C
                286*      C  NO MORE DATA ON DISK, CK FOR A UNPROCESSED BUFFER OF DATA FROM A
                287*      C  LOOK AHEAD READ
                288*        183 BUF=BUF+1
                289*            IF(BUF .EQ. 2)GO TO 200
                290*      C
                291*        184 IF(ISTAT(BN) .EQ. 1) GO TO 184
                292*      C
                293*      C  JMP = 1  FIELD COMPLETELY OFF THIS PAGE
                294*      C  JMP = 2  IF MORE WORDS IN BUFFER THAN EXTRANEOUS POINTS
                295*      C  JMP = 3  IF NO. WORDS IN BUFFER = NO PTS PROCESSED + EXTRANEOUS PTS
                296*      C  JMP = 4  IF MORE EXTRANEOUS PTS THAN WORDS IN BUFFER
                297*        185 GO TO(190,165,167,167),JMP
                298*      C
                299*        190 BUFCT(BN)=BUFCT(BN)+TNPTS
                300*            BUFAD=BUFCT(BN)+1
                301*      C
                302*        200 CONTINUE
                303*      C
                304*        300 CONTINUE
                305*      C
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00505     306*      C                    ** END OF GENERATION OF LINES FOR 1 PAGE **                    001013
00505     307*      C                                                                                   001013
00505     308*      C  CHECK FOR ADDITIONAL PAGES                                                       001013
00507     309*            IF(.NOT. NFIN)GO TO 400                                                       001013
00507     310*      C                                                                                   001013
00507     311*      C  MULTIPLE PAGES, RESET BOUNDARIES                                                 001013
00511     312*            XTRA=(ENCLM-SAMSTR)/SAMINC + 1                                                001015
00512     313*            STCLM=ENCLM+1                                                                 001023
00513     314*            ENCLM=SAMEND                                                                  001026
00514     315*            NFIN=.FALSE.                                                                  001030
00514     316*      C                                                                                   001030
00514     317*      C  REWIND INPUT TAPE BY RESETTING THE DISK ADDRESS                                  001030
00515     318*            ADRESS(I)=ADFLD                                                               001031
00516     319*            TWRD=TOTWRD-WRDSV                                                             001033
00517     320*            WRDCT=WRDSV                                                                   001036
00517     321*      C                                                                                   001036
00517     322*      C  READ THE FIRST 2 BUFFERS OF DATA                                                 001036
00520     323*            CALL BFINIT                                                                   001040
00521     324*            GO TO 50                                                                      001042
00521     325*      C                                                                                   001042
00522     326*        400 CONTINUE                                                                      001044
00522     327*      C                                                                                   001044
00522     328*      C               **  END OF CLUSTER MAP **                                           001044
00522     329*      C                                                                                   001044
00522     330*      C                 ** PRINT COUNTS **                                                001044
00522     331*      C                                                                                   001044
00523     332*            ADFLD=ADFLD+(TOTSAM*NOFEAT)                                                   001044
00524     333*            WRDSV=WRDSV+(TOTSAM*NOFEAT)                                                   001047
00524     334*      C                                                                                   001047
00525     335*            DO 465 I=1,MAXPOP                                                             001056
00530     336*        465 NBLKT(I)=NBLKT(I)+NBLK(I)                                                     001056
00530     337*      C                                                                                   001056
00532     338*            WRITE(6,570)                                                                  001061
00534     339*        570 FORMAT(//2X,'POINTS PER CLUSTER IN THIS FIELD'/3X,'CLUSTER',                  001066
00534     340*           * 5X,'SYMBOL',5X,'POINTS'/)                                                    001066
00534     341*      C                                                                                   001066
00535     342*            LNCAT=MOD(LNCAT-1,MAXPOP)+1                                                   001066
00535     343*      C                                                                                   001066
00535     344*      C                                                                                   001066
00536     345*            DO 580 I=1,LNCAT                                                              001074
00541     346*        580 WRITE(6,590)I,SYM(I),NBLK(I)                                                  001100
00547     347*        590 FORMAT(6X,I2,10X,A1,7X,I5)                                                    001112
00547     348*      C                                                                                   001112
00550     349*            IPT=IPT+9+NV*2                                                                001112
00551     350*        600 CONTINUE                                                                      001122
00551     351*      C                                                                                   001122
00551     352*      C                                                                                   001122
00552     353*            IF(BUF .NE. 2) WRITE(6,630)                                                   001122
00556     354*        630 FORMAT(' ALL OF DATA NOT READ---PROBLEM IN CLUSMP')                           001132
00556     355*      C                                                                                   001132
00557     356*            WRITE(6,HEAD)                                                                 001132
00561     357*            WRITE(6,750)LNCAT                                                             001137
00564     358*        750 FORMAT(//' TOTAL NUMBER OF CLUSTERS =',I3)                                    001145
00564     359*      C                                                                                   001145
00565     360*            TOTPTS=TOTWRD/NOFEAT                                                          001145
00565     361*      C                                                                                   001145
00566     362*            WRITE(6,760) TOTPTS                                                           001151
```

```
00571        760 FORMAT(/' TOTAL NUMBER OF POINTS =',I5)                         001161
00571      C                                                                     001161
00572            WRITE(6,770)                                                    001161
00574        770 FORMAT(//' CLUSTER     SYMBOL     POINTS IN CLUSTER')           001171
00574      C                                                                     001171
00575            DO 775 J=1,LNCAT                                                001171
00600        775 WRITE(6,780) J,SYM(J),NBLKT(J)                                  001171
00606        780 FORMAT(4X,I2,9X,A1,10X,I7)                                      001203
00606      C                                                                     001203
00607            RETURN                                                          001203
00607      C*                                                                    001203
00607      C*    INTERNAL ROUTINE BFINIT                                         001203
00607      C*    BFINIT INITIALIZES 2 BUFFERS-  AT THE FIRST OF THE CLUSMP       001203
00607      C*    SUBROUTINE AND WHENEVER A CLUSTER MAP EXCEEDS 110 SYMBOLS ACROSS 001203
00607      C*    AND THE DATA FOR THAT MAP MUST BE READ MORE THAN ONCE           001203
00607      C*                                                                    001203
00610            SUBROUTINE BFINIT                                               001244
00610      C                                                                     001244
00610      C  INPUT                                                              001244
00610      C    TWRD = TOTAL WORDS LEFT TO READ                                  001244
00610      C    ADRESS(1) = ADDRESS FOR 1ST READ                                 001244
00610      C  OUTPUT                                                             001244
00610      C  BUF = 1IF 1 BUFFER,  0 IF 2 BUFFERS                                001244
00610      C    BN = 1                                                           001244
00610      C    NOWRD(1),(2) = NO. OF WORDS IN BUFFERS                           001244
00610      C    WRDCT = TOTAL WORDS READ                                         001244
00610      C    BUFCT(1),(2) = 0                                                 001244
00610      C    BUFAD = 1                                                        001244
00610      C    ADDRESS = ADDRESS FOR 2ND READ                                   001244
00610      C                                                                     001244
00610      C                                                                     001244
00613            BUF=0                                                           001244
00613      C*    SET BUFER NUMBER                                                001244
00614            BN=1                                                            001245
00615            NOWRD(1)=MXBF /NOFEAT * NOFEAT                                  001247
00616            IF(TWRD .LT. NOWRD(BN)) NOWRD(BN)=TWRD                          001256
00620            NOWRD(2)=NOWRD(1)                                               001265
00621            WRDCT=WRDCT+NOWRD(BN)                                           001267
00622            BUFCT(1)=0                                                      001273
00623            BUFCT(2)=0                                                      001274
00624            BUFAD=1                                                         001275
00625            CALL RREAD(ADRESS(1),BUFER(1,1),NOWRD(1),ISTAT(1))              001277
00626            IF(WRDCT .NE. TOTWRD)GO TO 1110                                 001305
00630            BUF=1                                                           001310
00631            GO TO 1120                                                      001312
00632       1110 ADRESS(2)=ADRESS(1)+NOWRD(1)                                    001314
00633            IF(WRDCT + NOWRD(2) .LE. TOTWRD)GO TO 1115                      001316
00635            NOWRD(2)=TOTWRD-WRDCT                                           001323
00636       1115 WRDCT=WRDCT + NOWRD(2)                                          001327
00637            CALL RREAD(ADRESS(2),BUFER(1,2),NOWRD(2),ISTAT(2))              001331
00640       1120 IF(ISTAT(1) .EQ. 1) GO TO 1120                                  001340
00642            RETURN                                                          001342
00643            END                                                             001355

END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S CORECT,CORECT,CORECT
FOR (3E3-04/16/78-01:28:35 ID,O)

   SUBROUTINE CORECT    ENTRY POINT 000C32

   STORAGE USED: CODE(1) 000050; DATA(0) 000017; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE   ATION, NAME)

    C001     000012 1336      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX
    C004     000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER    0004    000102 BIAS
    C003 R   000027 CIN       0005    000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT      0003 R  000041 DCON
    C004     000053 DELT      0003 R  000043 DISS      0004    000073 DWFAC     0004    000051 ELIMTH    0004    000052 EPS
    C003 R   000050 GEN       0004    000074 GRACTM    0003 R  000047 GREF      0000 I  000000 I         0003 I  000023 IDADJ
    C003 I   000024 INDEX     0004    000066 INDXVL    0000    000002 INJP$     0003 I  000030 JUNK      0004    000051 KROOT
    C003 I   000020 LINK      0004    000003 LKURT     0004    000007 LOSUM     0004    000034 LOVAR     0004 I  000002 LR
    C004     000006 LSKEW     0007 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV
    C004     000002 LVRIN     0005 R  000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I  000000 MQ
    C004 I   000011 MXAR      0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO     0003 I  000043 NSYMB
    C003 I   000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT     0003 R  000045 OCIN
    C004     000055 ODCOR     0003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW
    C005 R   000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS
    C003 R   000042 PQRAT     0004    000071 PQRATH    0003 R  000026 PRIRCM    0003 R  000031 PROP      0003 R  000044 PST
    C004     000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR
    C003 R   000032 SPFAC     0004    000072 SPMVTH    0005    000004 TPBND     0005    000005 TRCHI     0005    000006 URKBND
    C005     000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN     0004    000101 VOLLIM
    C003 R   000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R  000034 W         0003 R  000033 WADJ
    C004     000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIM
    C004     000058 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*            SUBROUTINE CORECT(REL,PV,P,S)                                          000012
00101       2*      C  THIS ROUTINE CALCULATES AN UNWEIGHTED NORMALIZED VECTOR.                  000012
00103       3*            INCLUDE MISH                                                           000012
00131       4*            REAL REL(30),PV(30),S(30)                                              000012
00132       5*            DO 10 I=1,MQ                                                           000012
00135       6*       10 REL(I)=PV(I)-S(I)/P                                                      000012
00137       7*            RETURN                                                                 000016
00140(      8*            END                                                                    000047
```

```
END OF COMPILATION:  1 DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S DATFIX,DATFIX,DATFIX
FOR SDE3-04/18/78-01:28:38 (2,0)

   SUBROUTINE DATFIX    ENTRY POINT 000035

   STORAGE USED: CODE(1) 000041; DATA(0) 000016; BLANK COMMON(2) 000000

    (COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016
    0006    SPPAR   000022
    0007    INITL   000003
    0010    JOINPR  000004

   EXTERNAL REFERENCES (BLOCK, NAME)

    0011    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000022 145G      0003 R 000020 ALINN     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX
    0004    000076 AMOMIN    0004   000100 AMORAT    0004    000054 AMO       0004    000107 BETTER    0004    000102 BIAS
    0007    000002 CHANIN    0003 R 000027 CIN       0005 R  000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT
    0006    000013 DAMP      0003 R 000041 DCON      0006    000001 DELMET    0004    000053 DELT      0007    000001 DEVINI
    0006    000015 DIAG      0003 R 000043 DIS$      0006    000014 DOBPMS    0004    000073 DWFAC     0004    000061 ELIMTH
    0004    000052 EPS       0006   000010 EXMAX     0006    000006 EXMNSQ    0006    000011 GAMCEN    0006    000000 GAMMET
    0003 R  000050 GEN       0004   000074 GRACTM    0003 R  000047 GREF      0000 I  000000 I         0003 I  000023 IDADJ
    0003 I  000024 INDEX     0004   000066 INDXVL    0000    000010 INJPS     0006    000020 ITERMX    0003 I  000000 JUNK
    0004    000051 KROOT     0003 I 000020 LINK      0004    000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR
    0004 I  000002 LR        0004   000006 LSKEW     0003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER
    0004 I  000005 LV        0004   000002 LVRIN     0005 R  000012 MACCEL    0004    000001 MM        0004 I  000110 MODE
    0004    000000 MQ        0004 I 000011 MXAR      0003    000014 NARL      0004    000010 NINCLS    0010    000003 NOELIM
    0010    000002 NOJO      0004   000070 NPTSO     0003 I  000043 NSYMB     0003 I  000020 NTB       0003    000016 NTBSZM
    0003    000015 NTOP      0003   000017 NWANT     0006    000003 OBCOV     0006    000005 OBKURT    0006    000004 OBSKEW
    0003 P  000045 OCIN      0004   000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROJ
    0003 R  000036 OW        0005 R 000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN
    0003 P  000043 PPASS     0003 R 000042 PQRAT     0004    000071 PQRATH    0003 R  000026 PRIRCM    0003 R  000031 PROP
    0003 R  000044 PST       0010   000001 RLIM      0004    000065 SBLTH     0004    000062 SEPTH     0006    000002 SGTMET
    0006    000007 SHRMIN    0005   000002 SKBND     0005    000003 SKCHI     0004 R  000112 SPCOR     0003 R  000032 SPFAC
    0004    000072 SPMVTH    0006   000021 SPRED     0006    000016 TIMO      0006    000017 TIM1      0005    000004 TRBND
    0005    000005 TRCHI     0006   000012 TSQINI    0005    000006 URKBND    0005    000007 URKCHI    0005 R  000014 VACCEL
    0004    000063 VFAC      0003 R 000037 VOLIN     0004    000101 VOLLIM    0003 R  000040 VOLRT     0003 R  000050 VRIN
    0004    000104 VRJOIN    0003 R 000034 W         0003 R  000033 WADJ      0004 R  000060 WADJIN    0005 R  000000 WAIT
    0004 R  000106 WDELSM    0010   000000 WDJOIN    0004 R  000067 WFAC      0004 R  000105 WSIM      0004    000050 WTINIT
    0007    000000 WTNEW     0004   000056 XOVFLO    0004    000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            SUBROUTINE DATFIX                                                          000000
```

```
00101          INCLUDE MISH                                                      000000
00131          COMMON/SPPAR/ GAMMET,DELMET,SGTMET,OBCOV,OBSKEW,OBKURT,EXMNSQ,    000000
00131         1   SHRMIN,EXMAX,GAMCEN,TSOINI,DAMP,DOBPMS,DIAG,TIMO,TIM1,ITERMX,  000000
00131         2    SPREC                                                         000000
00132          COMMON /INITL/WTNEW,DEVINI,CHANIN                                 000000
00133          COMMON /JOINPR/WDJOIN,RLIM,NOJO,NOELIM                            000000
00134          MODE=0                                                            000000
00135           CONLV=3.                                                         000000
00136           WFAC=10.                                                         000002
00137          WAIT=150.                                                         000004
00140          WDELSM=20.                                                        000006
00141          WSIM=400.                                                         000010
00142          WADJIN=200.                                                       000012
00143          CONLV=CONLV*SPCOR                                                 000014
00144          DO 20 I=1,NTBSIZ                                                  000022
00147       20 NTB(I)=0                                                          000022
00151          RETURN                                                            000023
00152          END                                                               000040

     END OF COMPILATION:          1 DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S DENCAL,DENCAL,DENCAL
FOR SOE3-C4/18/78-01:28:42 (0,0)

   SUBROUTINE DENCAL    ENTRY POINT 000033

   STORAGE USED: CODE(1) 000037; DATA(0) 000010; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0003 R 000020 ALINK     0004   000064 AMM      0004   000075 AMOFAC   0004   000077 AMOMAX   0004   000076 AMOMIN
    0004   000100 AMORAT    0004   000054 AMQ      0004   000107 BETTER   0004   000102 BIAS     0003 R 000027 CIN
    0005   000001 CONLV     0004   000111 CORLEN   0003 R 000030 CTOT     0003 R 000041 DCON     0004   000053 DELT
    0003 R 000043 DISS      0004   000073 DWFAC    0004   000061 ELIMTH   0004   000052 EPS      0003 R 000050 GEN
    0004   000074 GRACTM    0003 R 000047 GREF     0003 I 000023 IDADJ    0003 I 000024 INDEX    0004   000066 INDXVL
    0000   000002 INJPS     0003 I 000000 JUNK     0000 I 000000 KF       0004   000051 KROOT    0003 I 000020 LINK
    0004   000003 LMURT     0004   000007 LOSUM    0004   000004 LOVAR    0004 I 000002 LR       0004   000006 LSKEW
    0003 I 000021 LSUBS     0004   000005 LSUM     0003 I 000022 LSUPER   0004 I 000005 LV       0004   000002 LVRIN
    0005 R 000012 MACCEL    0004   000001 MM       0004   000110 MODE     0004   000000 MQ       0004 I 000011 MXAR
    0003   000014 NARL      0004   000010 NINCLS   0004   000070 NPTSO    0003 I 000043 NSYMB    0003 I 000020 NTB
    0003   000016 NTBSZM    0003   000015 NTOP     0003   000017 NWANT    0003 R 000045 OCIN     0004   000055 ODCON
    0003 R 000047 ODEN      0000 R 000001 OLF      0003 R 000046 OPRIOR   0003 R 000035 OPROP    0003 R 000036 OW
    0005 R 000010 PACCEL    0003 R 000050 PCOND    0003 R 000025 PCUM     0004   000103 PJOIN    0003 R 000043 PPASS
    0003 R 000042 PQRAT     0004   000071 PQRATH   0003 R 000026 PRIRCM   0003 R 000031 PROP     0003 R 000044 PSI
    0004   000065 SBLTH     0004   000062 SEPTH    0005   000002 SKBND    0005   000003 SKCHI    0004   000112 SPCOR
    0003 R 000032 SPFAC     0004   000072 SPMVTH   0005   000004 TRBND    0005   000005 TRCHI    0005   000006 URMBND
    0005   000007 URKCHI    0005 R 000014 VACCEL   0004   000063 VFAC     0003 R 000037 VOLIN    0004   000101 VOLLIM
    0003 R 000040 VOLRT     0003 R 000056 VRIN     0004   000104 VRJOIN   0003 R 000034 W        0003 R 000033 WADJ
    0004   000060 WADJIN    0005   000000 WAIT     0004   000106 WDELSM   0004   000067 WFAC     0004   000105 WSIM
    0004   000050 WTINIT    0004   000056 XOVFLO   0004   000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101        1*          SUBROUTINE DENCAL(KL,RATIO,OLW)                                          000000
00101        2*      C                                                                            000000
00101        3*      C                                                                            000000
00101        4*                                                                                   000000
00101        5*      C  THIS ROUTINE ADJUSTS THE DENOMINATOR OFFSET AND PROPORTION OF KL.         000000
00101        6*      C    NEW PROP=RATIO*OLD PROP      -----      OLW=OLD W(XPATH)              000000
00101        7*      C   THE NODES MUST ALREADY BE RECONNECTED TO THEIR NEW POSITIONS.            000000
00101        8*          INCLUDE MISH                                                             000000
00101        9*      C                                                                            000000
```

```
00101     10*      C                                                              000000
00131     11*           PROP(KL)=PROP(KL)*RATIO                                   000000
00132     12*           OPROP(KL)=OPROP(KL)*RATIO                                 000003
00133     13*           KF=LSUPER(KL)                                             000006
00134     14*           OLF=CTOT(KL)                                              000010
00135     15*           CTOT(KL)=W(KF)-(OLW-CTOT(KL))/RATIO                       000012
00136     16*           ODEN(KL)=ODEN(KL)/RATIO                                   000017
00139     17*           RETURN                                                    000022
00140     18*           END                                                       000036
```

END OF COMPILATION: 1 DIAGNOSTICS.

ORIGINAL PAGE IS
OF POOR QUALITY

```
BEGR,S DOTSQ,DOTSQ,DOTSQ
FOR SOE3-04/18/78-01:28:45 (0,0)

   FUNCTION DOTSQ        ENTRY POINT 000105

   STORAGE USED: CODE(1) 000123; DATA(0) 000030; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003   CLUS    002017
    C004   MISC    000113
    C005   STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006   MERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000022 1356       0001    000042 141G       0003 R 000020 ALINK      0004    000064 AMM        0004    000075 AMOFAC
    0004    000077 AMOMAX     0004    000076 AMOMIN     0004    000100 AMORAT     0004    000054 AMQ        0004    000107 BETTER
    0004    000102 BIAS       0003 R  000027 CIN        0005    000001 CONLV      0004    000111 CORLEN     0003 R  000030 CTOT
    0003 R  000041 DCON       0004    000053 DELT       0000 R  000001 DGDOT      0003 R  000043 DISS       0000 R  000000 DOTSQ
    0004    000073 DWFAC      0004    000061 ELIMTH     0004    000052 EPS        0003 R  00005G GEN        0004    000074 GRACTM
    0003 R  000047 GREF       0000 I  000002 I          0003 I  000023 IDADJ      0003 I  000024 INDEX      0004    000066 INDXVL
    0000    000006 INJPS      0000 I  000004 J          0003 I  000000 JUNK       0004    000051 KROOT      0003 I  000020 LINK
    0004    000003 LKURT      0004    000007 LOSUM      0004    000004 LOXAR      0004 I  000002 LR         0004    000006 LSKEW
    0002 I  000021 LSUBS      0004    000005 LSUM       0003 I  000022 LSUPER     0004 I  000005 LV         0004    000002 LVRIN
    0005 R  000012 MACCEL     0004    000001 MM         0004    000110 MODE       0004 I  000000 MQ         0000 I  000003 MX
    0004 I  000011 MXAR       0003    000014 NARL       0004    000010 NINCLS     0004    000070 NPTSQ      0003 I  000043 NSYMB
    0003    000020 NTB        0003    000016 NTBSZM     0003    000015 NTOP       0003    000017 NWANT      0003 R  000045 OCIN
    0004    000055 ODCON      0003 R  000047 ODEN       0003 R  000046 OPRIOR     0003 R  000035 OPROP      0003 R  000036 OW
    0005 R  000010 PACCEL     0003 R  000050 PCOND      0003 R  000025 PCUM       0004    000103 PJOIN      0003 R  000043 PPASS
    0003 R  000042 PORAT      0004    000071 POPATH     0003 R  000026 PRIRCM     0003 R  000031 PROP       0003 R  000044 PST
    0004    000065 SBLTH      0004    000062 SEPTH      0005    000002 SKBND      0005    000003 SKCHI      0004    000112 SPCOR
    0003 R  000032 SPFAC      0004    000072 SPMVTH     0005    000004 TRBND      0005    000005 TRCHI      0005    000006 URKBND
    0005    000007 URKCHI     0005 R  000010 VACCEL     0004    000063 VFAC       0003 R  000037 VOLIN      0004    000101 VOLLIM
    0003 R  000040 VOLRT      0003 R  000050 VRIN       0004    000104 VRJOIN     0003 R  000034 W          0003 R  000033 WADJ
    0004    000060 WADJIN     0005    000000 WAIT       0004    000106 WDELSM     0004    000067 WFAC       0004    000105 WSIM
    0004    000050 WTINIT     0004    000056 XOVFLO     0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00001       1*            FUNCTION DOTSQ(V,AMET)                                                    000007
00001       2*      C       CALCULATES THE INNER PRODUCT V.V RELATIVE TO THE METRIC AMET             000007
00001       3*            INCLUDE MISH                                                              000007
00001       4*            REAL V(3G),AMET(475)                                                      000007
00001       5*            DOTSQ=0.                                                                  000007
00001       6*            DGDOT=V(1)*V(1)*AMET(1)                                                   000010
00001       7*            DO 10 I=2,MQ                                                              000022
00001       8*            MX=MXAR(I)                                                                000025
```

```
00140       9*          7 DO 8 J=2,I                                                       000027
00143      10*          8 DOTSQ=DOTSQ+V(I)*V(J-1)*AMET(MX+J-1)                             000042
00145      11*         10 DGDOT=DGDOT+V(I)*V(I)*AMET(MX+I)                                 000047
00143      12*    C          THE DIAGONALS ARE HANDLED SEPARATELY BECAUSE EACH OFF-        000047
00145      13*    C          DIAGONAL APPEARS TWICE, AND SO MUST BE DOUBLED.               000047
00147      14*            DOTSQ=DOTSQ+DOTSQ+DGDOT                                          000057
00150      15*            RETURN                                                           000062
00151      16*            END                                                              000122

     END OF COMPILATION:          1 DIAGNOSTICS.
```

ORIGINAL PAGE IS OF POOR QUALITY

```
@FOR,S EIGROT,EIGROT,EIGROT
FOR 30E3-04/18/78-01:28:48 (0,0)

   SUBROUTINE EIGROT    ENTRY POINT 000135

   STORAGE USED: CODE(1) 000172; DATA(0) 000014; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    GET
    0007    TRIDMX
    0010    EIGVAL
    0011    EIGVEC
    0012    FREE
    0013    NERRJS

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0003 R 000020 ALINK      0004    000064 AMM        0004    000075 AMOFAC     0004    000077 AMOMAX     0004    000076 AMOMIN
    0004   000100 AMORAT     0004    000054 AMS        0004    000107 BETTER     0004    000102 BIAS       0003 R  000027 CIN
    0005   000001 CONLV      0004    000111 CORLEN     0003 R  000030 CTOT       0003 R  000041 DCON       0004    000053 DELT
    0003 R 000043 DISS       0004    000073 DWFAC      0004    000061 ELIMTH     0004    000052 EPS        0003 R  000050 GEN
    0006 I 000000 GET        0004    000074 GRACTM     0003 R  000047 GREF       0000 I  000001 IB         0000 I  000000 ID
    0003 I 000023 IDADJ      0000 I  000003 IF         0003 I  000024 INDEX      0004    000066 INDXVL     0000    000004 INJPS
    0000 I 000002 IW         0003 I  000000 JUNK       0004    000051 KROOT      0003 I  000020 LINK       0004    000003 LKURT
    0004   000007 LOSUM      0004    000004 LOVAR      0004 I  000002 LP         0004    000305 LSKEW      0003 I  000021 LSUBS
    0004   000005 LSUM       0003 I  000022 LSUPER     0004 I  000005 LV         0004    000002 LVRIN      0005 R  000012 MACCEL
    0004   000001 MM         0004    000110 MODE       0004    000000 MQ         0004 I  000011 MXAR       0003    000014 NARL
    0004   000010 NINCLS     0004    000070 NPTSO      0003 I  000043 NSYMB      0003 I  000020 NTB        0003    000016 NTBSZM
    0003   000015 NTOP       0003    000017 NWANT      0003 R  000045 OCIN       0004    000055 ODCON      0003 R  000047 ODEN
    0003 R 000046 OPRIOR     0003 R  000035 OPROP      0003 R  000036 OW         0005 R  000010 PACCEL     0003 R  000050 PCOND
    0003 R 000025 PCUM       0004    000103 PJOIN      0003 R  000043 PPASS      0003 R  000042 PQRAT      0004    000071 PQRATH
    0003 R 000026 PRIRCM     0003 R  000031 PROP       0003 R  000044 PST        0004    000065 SBLTH      0004    000052 SEPTH
    0005   000002 SKBND      0005    000003 SKCHI      0004    000112 SPCOR      0003 R  000032 SPFAC      0004    000072 SPMVTH
    0005   000004 TRBND      0005    000005 TRCHI      0005    000006 URKBND     0005    000007 URKCHI     0005 R  000014 VACCEL
    0004   000063 VFAC       0003 R  000037 VOLIN      0004    000101 VOLLIM     0003 R  000040 VOLRT      0003 R  000050 VRIN
    0004   000104 VRJOIN     0003 R  000034 W          0003 R  000033 WADJ       0004    000060 WADJIN     0005    000000 WAIT
    0004   000106 WDELS4     0004    000067 WFAC       0004    000105 WSIM       0004    000050 WTINIT     0004    000056 XOVFLO
    0004   000057 XUNFLO

00101      1*          SUBROUTINE EIGROT(LP,NM,R,E,V)                                                    000000
00101      2*      C  THIS ROUTINE CALLS SYSTEM ROUTINES TO GENERATE AN EIGENROTATION OF                  000000
00101      3*      C     AN LP*LP SUBMATRIX OF THE ARRAY R. THE EIGENVALUES ARE RETURNED                 000000
00101      4*      C     IN E AND THE EIGENVECTOR MATRIX IS IN V (DIM NM*NM), WHERE                       000000
```

```
00101      5*      C       THE SECOND INDEX RUNS OVER EIGENVECTORS, AND THE FIRST                  000000
00101      6*      C       WITHIN THEM.                                                            000000
00101      7*      C       THE STORAGE ALLOCATION SYSTEM (GET,FREE) IS ALSO USED.                  000000
00101      8*      C  THE LOWER TRIANGLE OF R IS DESTROYED.                                        000000
00101      9*              REAL R(LP,LP),E(LP),V(LP,LP)                                            000000
00104     10*              INCLUDE MISH                                                            000000
00132     11*              ID=GET(LP)                                                              000000
00133     12*              IB=GET(LP)                                                              000003
00134     13*              CALL TRIDMX(LP,NM,R,LINK(ID),LINK(IB))                                  000010
00135     14*              IW=GET(LP)                                                              000024
00136     15*              IF=GET(LP)                                                              000030
00137     16*              CALL EIGVAL(LP,E,LINK(ID),LINK(IB),LINK(IW),LINK(IF))                   000035
00137     17*      C       PRINT 666 , ID,IB,IW,IF,LP,NM,E                                         000035
00137     18*      C  666 FORMAT(' EIGPOT:',6I6,4 E16.7)                                           000035
00140     19*              CALL EIGVEC(LP,NM,R,LINK(ID),LINK(IB),E,V,LINK(IF),LINK(IW))            000056
00141     20*              CALL FREE(ID,LP)                                                        000101
00142     21*              CALL FREE(IB,LP)                                                        000105
00143     22*              CALL FREE(IW,LP)                                                        000111
00144     23*              CALL FREE(IF,LP)                                                        000115
00145     24*              RETURN                                                                  000121
00146     25*              END                                                                     000171

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

BEGINNING OF PAGE IS MAINLY NOT FILMED

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S ELIM,ELIM,ELIM
FOR SOE3-04/18/78-01:31:01 (0,0)

   SUBROUTINE ELIM      ENTRY POINT 000160

   STORAGE USED: CODE(1) 000176; DATA(0) 000046; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003    CLUS    002017
    [004    MISC    000113
    [005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006    SUBLIM
    [007    TRFREE
    [010    NPRT$
    [011    NIOZ$
    [012    NIO1$
    [013    MERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001    000132 13L      0001    000136 15L      0001    000070 5L       0000    000020 666F     0001    000075 7L
    [000    000000 719F     0003 R  000020 ALINK    0004    000064 AMM      0004    000075 AMOFAC   0004    000077 AMOMAX
    [004    000076 AMOMIN   0004    000100 AMORAT   0004    000054 AMQ      0004    000107 BETTER   0004    000102 BIAS
    [003 P  000027 CIN      0005    000001 CONLV    0004    000111 CORLEN   0003 R  000030 CTOT     0003 R  000041 DCON
    [004    000053 DELT     0003 R  000043 DISS     0004    000073 DWFAC    0004    000061 ELIMTH   0004    000052 EPS
    [003 P  000050 GEN      0004    000074 GRACTM   0003 R  000047 GREF     0003 I  000023 IDADJ    0003 I  000024 INDEX
    [004    000066 INDXVL   0000    000032 INJPS    0003 I  000000 JUNK     0000 I  000005 K        0000 I  000000 KF
    [000 I  000001 KMEX     0003 I  000006 KOLD     0004 I  000051 KPOOT    0003 I  000020 LINK     0000 I  000004 LKI
    [004    000003 LKURT    0004    000007 LOSUM    0004    000004 LOVAR    0004 I  000002 LR       0000 I  000002 LS
    [004    000006 LSKEW    0000 I  000003 LSS      0003 I  000021 LSUBS    0004    000005 LSUM     0003 I  000022 LSUPER
    [004 I  000005 LV       0004    000002 LVRIN    0005 R  000012 MACCEL   0004    000001 MM       0004    000110 MODE
    [004    000000 MQ       0004 I  000011 MXAR     0003    000014 NARL     0004 I  000010 NINCLS   0004    000070 NPTSO
    [003 I  000043 NSYMB    0003 I  000020 NTB      0003    000016 NTBSZM   0003    000015 NTOP     0003    000017 NWANT
    [003 P  000045 OCIN     0004    000055 ODCON    0003 R  000047 ODEN     0003 R  000046 OPRIOR   0003 R  000035 OPROP
    [003 P  000036 OW       0005 R  000010 PACCEL   0003 R  000050 PCOND    0003 R  000025 PCUM     0004    000103 PJOIN
    [003 P  000043 PPASS    0003 R  000042 PORAT    0004    000071 PORATH   0003 R  000026 PRIRCM   0003 R  000031 PROP
    [003 P  000044 PST      0004    000065 SBLTH    0004    000062 SEPTH    0005    000002 SKBND    0005    000003 SKCHI
    [004    000112 SPCOR    0003 R  000032 SPFAC    0004    000072 SPMVTH   0005    000004 TRBND    0005    000005 TRCHI
    [005    000006 UPKBND   0005    000007 URKCHI   0005 R  000014 VACCEL   0004    000063 VFAC     0003 R  000037 VOLIN
    [004    000101 VOLLIM   0003 R  000040 VOLRT    0003 R  000050 VRIN     0004    000104 VRJOIN   0003 R  000034 W
    [003 R  000033 WADJ     0004    000060 WADJIN   0005    000000 WAIT     0004    000106 WDELSM   0004    000067 WFAC
    [004    000135 WSIM     0004    000050 WTINIT   0004    000056 XOVFLO   0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*          SUBROUTINE ELIM(KEL)                                                    000000
00101       2*      C   THIS ROUTINE ELIMINATES THE CLUSTER KEL FROM THE CLUSTER TREE           000000
```

```
C      AND FREES THE STORAGE.
       INCLUDE MISH
C
C
C  KF=PARENT, KMEX=1ST SIB, LS = OFFSPRING
       KF=LSUPER(KEL)
       KMEX=LINK(KEL)
       LS=LSUBS(KEL)
       PRINT 719,INDEX(KEL),INDEX(KMEX),INDEX(LS),INDEX(KF)
  719 FORMAT('0***ELIMINATE',I4,'   LINK,LSUBS,LSUPER=',3I3)
C
C  FIRST, USE SUBLIM IF THERE ARE ONLY 2 SUBCLUSTERS AT THIS LEVEL.
       LSS=LSUBS(KF)
       IF(KF.EQ.KROOT.AND.LINK(LSS).EQ.0) RETURN
       LKI=LINK(LSS)
       IF(LINK(LKI).NE.0.OR.KF.EQ.KROOT) GO TO 5
     4 CALL SUBLIM(KF)
       RETURN
C
C
C   NOW WE REMOVE THE CLUSTER FROM VARIOUS LISTS.
C
     5 K=LSUBS(KF)
       IF(K.EQ.KEL) GO TO 13
C
C  KET NOT 1ST OFFSPRING
     7 KOLD=K
       K=LINK(K)
       IF(K.EQ.0) PRINT 666,KEL,KF,KOLD,LSUBS(KF)
  666 FORMAT('C **STRUCTURAL ERROR AT ELIM: KEL,KFATH,KOLD,INIT',5I9)
       IF(K.NE.KEL) GO TO 7
C
C  NODE KEL FOUND, AS NTH OFFSPRING (N NOT 1), SET LINK OF N-1 TO N+1
       LINK(KOLD)=LINK(K)
       GO TO 15
C
C  NODE KEL IS 1ST OFFSPRING, SET 1ST OFFSPRING LINK TO LINK FROM KEL
   13 LSUBS(KF)=LINK(K)
   15 LINK(KEL)=0
C
C   NOW DROP THE CLUSTER AND ITS SUBS
       CALL TRFREE(KEL,NINCLS)
       RETURN
       END
```

END OF COMPILATION: 1 DIAGNOSTICS.

```
@FOR,S ISPLIT,ISPLIT,ISPLIT
FOR SCE3:04/18/78-01:31:25 (0,0)

   FUNCTION ISPLIT      ENTRY POINT 000044

   STORAGE USED: CODE(1) 000052; DATA(0) 000015; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003    CLUS    002017
    [004    MISC    000113
    [005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [003  P 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX    0004    000076 AMOMIN
    [004    000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER    0004    000102 BIAS      0003  R 000027 CIN
    [005    000001 CONLV     0004    000111 CORLEN    0003  R 000030 CTOT      0003  R 000041 DCON      0004    000053 DELT
    [003  R 000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS       0003  R 000050 GEN
    [004    000074 GRACTM    0003  R 000047 GREF      0003  I 000023 IDADJ     0003  I 000024 INDEX     0004    000066 INDXVL
    [005    000003 INJP$     0000  L 000000 ISPLIT    0003  I 000000 JUNK      0000  I 000001 KL        0004    000051 KROOT
    [003  I 000020 LINK      0004    000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR     0004  I 000002 LR
    [000  I 000002 LSB       0004    000006 LSKEW     0003  I 000021 LSUBS     0004    000005 LSUM      0003  I 000022 LSUPER
    [004  I 000005 LV        0004    000002 LVRIN     0005  R 000012 MACCEL    0004    000001 MM        0004    000110 MODE
    [004    000030 MQ        0004  I 000011 MXAR      0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO
    [003  I 000043 NSYMB     0003  I 000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT
    [003  P 000045 OCIN      0004    000055 ODCON     0003  R 000047 ODEN      0003  R 000046 OPRIOR    0003  R 000035 OPROP
    [003  P 000036 OW        0005  R 000010 PACCEL    0003  R 000050 PCOND    0003  R 000025 PCUM      0004    000103 PJOIN
    [003  P 000043 PPASS     0003  R 000042 PQRAT     0004    000071 PQRATH    0003  R 000026 PRIRCM    0003  R 000031 PROP
    [003  P 000044 PST       0004    000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI
    [004    000112 SPCOR     0003  P 000032 SPFAC     0004    000072 SPMVTH    0005    000004 TRBND     0005    000005 TRCHI
    [005    000006 URKBND    0005    000007 URKCHI    0005  R 000014 VACCEL    0004    000053 VFAC      0003  R 000037 VOLIN
    [004    000101 VOLLIM    0003  R 000040 VOLRT     0003  R 000050 VRIN      0004    000134 VRJOIN    0003  R 000034 W
    [003  R 000033 WADJ      0004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC
    [004    000105 WSIM      0004    000050 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

000000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00010 1      1*          LOGICAL FUNCTION ISPLIT(KLI)                                                    000000
00010 1      2*          INCLUDE MISH                                                                    000000
00013 1      3*          KL=KLI                                                                          000000
00013 2      4*          LSB=LSUBS(KL)                                                                   000001
00013 3      5*          ISPLIT=LSB.NE.0.AND.(SPFAC(KL).GT.0..OR.                                        000004
00013 5      6*         1    IABS(INDEX(LSB)).LT.IABS(INDEX(KL)))                                        000004
00134        7*          RETURN                                                                          000030
00135        8*          END                                                                             000051
```

ORIGINAL PAGE IS OF POOR QUALITY

```
END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S JOIN,JOIN,JOIN
FOR S0E3-04/18/78-01:31:28 (0,0)

   FUNCTION JOIN         ENTRY POINT 000703

   STORAGE USED: CODE(1) 000764; DATA(0) 000125; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016
    0006    CLUSTR  000131

   EXTERNAL REFERENCES (BLOCK, NAME)

    0007    GET
    0010    SQMTX
    0011    MINV
    0012    APRIOR
    0013    DENCAL
    0014    TRIMTX
    0015    CLPR
    0016    NPRTS
    0017    NIO1S
    0020    NIO2S
    0021    XPRI
    0022    SQRT
    0023    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000444 244G      0001    000470 253G      0001    000560 264G      0001    000130 30L       0001    000151 35L
    0000    000032 653F      0000    000021 717F      0003 R  000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004 R  000054 AMQ       0012 R  000000 APRIOR
    0006    000123 AREA      0004    000107 BETTER    0004    000102 BIAS      0006 I  000115 BUFPIX    0006 I  000116 BUFTOT
    0000 R  000013 CA        0000 R  000014 CB        0000 R  000015 CBV       0000 R  000011 CF        0003 R  000027 CIN
    0006 I  000002 CLSNAM    0005    000001 CONLV     0004    000111 CORLEN    0003 R  000130 CTOT      0003 R  000041 DCON
    0003 R  000010 DD        0004    000053 DELT      0000 R  000017 DELTA     0003 R  000143 DISS      0006    000113 DUMP
    0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS       0000 R  000012 FA        0003 R  000050 GEN
    0007 I  000000 GET       0004    000074 GRACTM    0003 R  000047 GPEF      0000 I  000116 I         0006    000000 IBEGIN
    0003 I  000023 IDADJ     0003 I  000024 INDEX     0004 I  000066 INDXVL    0000    000152 INJPS     0006    000003 IPT
    0006    000130 IQ1       0000 I  000020 J         0000 I  000000 JOIN      0003 I  000100 JUNK      0000 I  000006 K
    0000 I  000001 KA        0003 I  000003 KB        0006    000107 KLBC      0000 I  000107 KO        0004    000051 KROOT
    0006    000127 LBUF      0006    000121 LBUFD     0000 I  000004 LIKKA     0003 I  000020 LINK      0004    000003 LKURT
    0006    000102 LNCAT     0004    000007 LOSUM     0004    000004 LOVAR     0004 I  000002 LR        0000 I  000002 LS
    0004    000006 LSKEW     0003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0000 I  000005 LSUPKA
    0004 I  000005 LV        0004    000002 LVRIN     0005 R  000012 MACCEL    0006    000122 MAXBF     0004 I  000001 MM
    0004    000110 MODE      0004 I  000000 MQ        0004 I  000011 MXAR      0003    000114 NARL      0006    000117 NBUFSD
    0006    000120 NDUMP     0004 I  000010 NINCLS    0006    000106 NIT       0006    000104 NOFLD     0006    000126 NPTS
    0004 I  000070 NPTS0     0003 I  000043 NSYMB     0003 I  000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP
    0003    000017 NWANT     0006    000125 NWDRS     0006    000124 NWDS      0003 R  000145 OCIN      0004 R  000055 ODCON
    0003 P  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000136 OW        0005 R  000010 PACCEL
```

```
0003 R 000050 PCOND      0003 R 000025 PCUM      0004   000103 PJOIN      0003 R 000043 PPASS      0003 R 000042 PRAT
0004   000071 PRATH      0003 R 000026 PRIRCM    0006 I 000103 PRNT       0003 R 000031 PROP       0006 I 000111 PROUT
0006 I 000110 PRTME      0003 R 000044 PST       0004   000065 SELTH      0006 I 000114 SCRAM1     0004   000062 SEPTH
0005   000002 SKBND      0005   000003 SKCHI     0004 R 000112 SPCOR      0003 R 000032 SPFAC      0004   000072 SPMVTH
0006 I 000005 SYM        0006 I 000112 TOTPIX    0006 I 000001 TOTWRD     0005   000004 TRBND      0005   000005 TRCHI
0005   000006 URKBND     0005   000007 URKCHI    0005 R 000014 VACCEL     0004   000063 VFAC       0003 R 000037 VOLIN
0004   000101 VOLLIM     0003 R 000040 VOLRT     0003 R 000050 VRIN       0004   000104 VRJOIN     0003 R 000034 W
0003 R 000033 WADJ       0004 R 000060 WADJIN    0005   000000 WAIT       0004   000106 WDELSM     0004 R 000067 WFAC
0004   000105 WSIM       0004   000050 WTINIT    0004   000056 XOVFLO     0004   000057 XUNFLO
```

```
00101     1*          INTEGER FUNCTION JOIN(KAI,KBI,SUM,SKEW,KURT,OSUM,OVAR,VVV,B,A,D)                    000024
00101     2*    C                                                                                         000024
00101     3*    C  JOIN  RAISES THE HYPOTHESIS THAT KA AND KB ARE THE SAME CLUSTER.                       000024
00101     4*    C    KB MUST BE OBTAINABLE FROM KA VIA LINK.                                              000024
00101     5*    C                                                                                         000024
00101     6*    C  CREATE NEW CLUSTER -JOIN- WITH KA AND KB AS SUBCLUSTERS                                000024
00101     7*    C                                                                                         000024
00101     8*    C  PURPOSE--                                                                              000024
00101     9*    C                                                                                         000024
00101    10*    C                                                                                         000024
00101    11*    C  WARNING: CALLING SUBROUTINE MUST ASSURE THAT KB IS TO RIGHT OF KA                      000024
00101    12*    C  *****  *****   *****   *****   *****   *****   *****   *****  *****                   000024
00101    13*    C                                                                                         000024
00151    14*    C                                                                                         000024
00151    15*          INCLUDE MISH                                                                        000024
00131    16*          INCLUDE CMBK10                                                                      000024
00131    17*    C                                                                                   CH    000024
00140    18*          REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)                                         000024
00141    19*          REAL A(MQ,MQ),B(MQ,MQ),D(MQ,MQ)                                                     000024
00142    20*          REAL VVV(MQ,MQ)                                                                     000024
00142    21*    C                                                                                         000024
00142    22*    C                                                                                         000024
00142    23*    C                    *** FIND CLUSTERS KA AND KB,                                         000024
00142    24*    C                        MOVE THEM FROM OLD STRING TO NEWLY CREATED                       000024
00142    25*    C                        STRING CONTAINING ONLY KA AND KB ***                             000024
00142    26*    C                                                                                         000024
00142    27*    C                                                                                         000024
00142    28*    C  KA      = GIVEN CLUSTER                                                                000024
00142    29*    C  KB      = GIVEN CLUSTER                                                                000024
00143    30*          KA=KAI                                                                              000024
00144    31*          LS=LSUPER(KA)                                                                       000027
00145    32*          KB=KBI                                                                              000032
00146    33*          JOIN=0                                                                              000035
00146    34*    C                                                                                         000035
00146    35*    C  LIKKA  = NEXT SIBLING OF KA                                                            000035
00146    36*    C  LSUPKA = PARENT OF KA                                                                  000035
00147    37*          LIKKA=LINK(KA)                                                                      000036
00150    38*          LSUPKA=LSUPER(KA)                                                                   000041
00150    39*    C                                                                                   CH    000041
00150    40*    C  LSUBS(LSUPKA) = 1ST OFFSPRING OF PARENT OF KA                                          000041
00150    41*    C  LINK(LIKKA)   = SIB OF SIB OF KA                                                       000041
00150    42*    C  CK FOR CLUSTER THAT IS NOT CONNECTED TO PARENT, PROB UNNECESSARY                       000041
00151    43*          IF(LSUBS(LSUPKA).EQ.0.AND.LINK(LIKKA).EQ.0) RETURN                                  000050
00151    44*    C                                                                                   CH    000050
```


ORIGINAL PAGE IS OF POOR QUALITY

```
C                                                                         CH      000050
C  JOIN = NEW CLUSTER                                                             000050
      JOIN=GET(NINCLS)                                                            000065
C                                                                                 000065
C  CREATE NEW CLUSTER -JOIN-                                                      000065
      INDXVL=INDXVL+1                                                             000071
      INDEX(JOIN)=INDXVL                                                          000075
      IDADJ(JOIN)=NPTSO+TOTPIX                                                    000076
      PRINT 717,INDEX(KA),INDEX(KB),INDEX(JOIN)                                   000101
  717 FORMAT(' C### JOINING  ',I4,' AND',I4,' TO GET',I4)                         000114
C                                                                                 000114
C  SET PARENT OF JOIN FROM PARENT OF KA                                           000114
C  SET LINK OF JOIN TO 1ST OFFSPRING OF PARENT                                    000114
C  SET JOIN TO BE 1ST OFFSPRING                                                   000114
C  SET OFFSPRING OF JOIN TO BE KA                                                 000114
      LSUPER(JOIN)=LS                                                             000114
      LINK(JOIN)=LSUBS(LS)                                                        000116
      LSUBS(LS)=JOIN                                                              000121
      LSUBS(JOIN)=KA                                                              000123
C                                                                         CH      000123
C  LINK DOWN SIBS OF JOIN TO KA                                                   000123
      K=JOIN                                                                      000125
   30 KO=K                                                                        000130
      K=LINK(K)                                                                   000133
      IF(K.NE.KA) GO TO 30                                                        000135
C                                                                                 000135
C  REMOVE KA FROM OLD FAMILY                                                      000135
C  RESET SIB POINTER FOR ELEMENT LINKING TO KA TO POINT TO LINK FROM KA           000135
C  SET SIB POINTER OF KA TO POINT TO KB                                           000135
C  SET UP PARENT POINTER OF KA TO BE JOIN                                         000135
      LINK(KO)=LINK(KA)                                                           000137
      LINK(KA)=KB                                                                 000142
      LSUPER(KA)=JOIN                                                             000144
C                                                                                 000144
C  LINK FROM KA TO KB--PROGRAM WILL ABORT IS KA DOES NOT PRECEDE KB               000144
      K=KO                                                                        000146
   35 KO=K                                                                        000151
      K=LINK(K)                                                                   000157
      IF(K.NE.KB) GO TO 35                                                        000161
C                                                                                 000161
C  RESET SIB POINTER THAT POINTS TO KB TO POINT TO SIB OF KB                      000161
      LINK(KO)=LINK(KB)                                                           000163
C                                                                         CH      000163
C  SET KB TO POINT TO 0                                                           000163
C  SET PARENT OF KB TO BE JOIN                                                    000163
      LINK(KB)=0                                                                  000166
      LSUPER(KB)=JOIN                                                             000167
C                                                                                 000167
C                                                                                 000167
C               *** CALCULATE STATISTICS FOR NEW CLUSTER ***                      000167
C                                                                                 000167
      CALL SQMTX(VVV,VRIN(KB+1))                                                  000171
      CALL MINV(B,D,VVV,DD)                                                       000177
C                                                                         CH      000177
      CALL SQMTX(VVV,VRIN(KA+1))                                                  000205
C                                                                         CH      000205
      CALL MINV(A,D,VVV,DD)                                                       000213
```

```
00214     102*      C
00214     103*      C   GET COVARIANCES OF THE PARTS.                                                    000213
00214     104*      C   CALCULATE INITIAL WEIGHTS                                                        000213
00215     105*            W(JOIN)=WFAC*AMQ*SPCOR                                                       000213
00216     106*            OW(JOIN)=W(JOIN)                                                             000221
00217     107*            WADJ(JOIN)=W(JOIN)*WADJIN                                                    000225
00217     108*      C                                                                                  000226
00217     109*      C   CALCULATE SPLITTING FACTORS                                                   000226
00220     110*            SPFAC(JOIN)=APRIOR(JOIN)                                                     000231
00221     111*            OPRIOR(JOIN)=SPFAC(JOIN)                                                     000235
00221     112*      C                                                                                  000235
00221     113*      C   CALCULATE PROPORTIONS FOR PARENT(JOIN) AND SUBS (KA + KB)                       000235
00222     114*            PORAT(JOIN)=0.                                                               000240
00223     115*            PROP(JOIN)=PROP(KA)+PROP(KB)                                                 000241
00224     116*            OPROP(JOIN)=PROP(JOIN)                                                       000244
00225     117*            PRIRCM(JOIN)=1.                                                              000245
00226     118*            CALL DENCAL(KA,1./PROP(JOIN),W(LS))                                          000247
00227     119*            CALL DENCAL(KB,1./PROP(JOIN),W(LS))                                          000261
00230     120*            CIN(JOIN)=CIN(KA)*PROP(KA)+CIN(KB)*PROP(KB)                                  000273
00231     121*            OCIN(JOIN)=CIN(JOIN)                                                         000303
00232     122*            ODEN(JOIN)=CIN(JOIN)/PROP(JOIN)                                              000304
00233     123*            CTOT(JOIN)=W(LS)-ODEN(JOIN)                                                  000307
00233     124*      C                                                                                  000307
00233     125*      C   CALCULATE WEIGHTING COEFFICIENTS (TEMPORARY-FOR MEANS AND COVAR)              000307
00233     126*      C                                                                                  000307
00234     127*            CF=W(JOIN)/(W(KB)*W(KB))*PROP(KA)*PROP(KB)                                   000311
00235     128*            FA=W(KB)/W(KA)                                                               000320
00236     129*            CA=PROP(KA)*W(JOIN)/W(KA)                                                    000323
00237     130*            CB=PROP(KB)*W(JOIN)/W(KB)                                                    000327
00240     131*            CBV=CB                                                                       000333
00241     132*            IF(INDEX(KB).LT.0) CBV=CBV*W(KB)/OW(KB)                                      000334
00241     133*      C   CALCULATE WEIGHTED OVERALL MEANS AND COVARIANCE                                000334
00243     134*            DO 21 I=1,MQ                                                                 000400
00246     135*            SUM(JOIN+I)=CA*SUM(KA+I)+CB*SUM(KB+I)                                        000444
00247     136*            SKEW(JOIN+I)=0.                                                              000451
00250     137*            OSUM(JOIN+I)=SUM(JOIN+I)                                                     000452
00251     138*            DELTA=CF*(FA*SUM(KA+I)-SUM(KB+I))                                            000455
00251     139*      C                                                                                  000455
00251     140*      C   COVARIANCE=COVAR(KA)+COVAR(KB)+DISPLACEMENT**2 (WITH COEFFICIENTS)             000455
00252     141*            DO 21 J=1,MQ                                                                 000470
00255     142*         21 D(I,J)=CA*A(I,J)+CBV*B(I,J)+DELTA*(FA*SUM(KA+J)-SUM(KB+J))                   000470
00255     143*      C                                                                                  000470
00255     144*      C   PUT COVARIANCE INTO JOIN NODE, CALCULATE VOLUME                          CH    000470
00260     145*            CALL TRIMTX(OVAR(JOIN+1),D)                                                  000524
00261     146*            CALL MINV(VVV,A,D,VOLIN(JOIN))                                               000533
00262     147*            CALL TRIMTX(VRIN(JOIN+1),VVV)                                                000544
00262     148*      C                                                                                  000544
00262     149*      C   ZERO OUT KURT                                                                  000544
00263     150*            DO 22 I=1,MM                                                                 000552
00266     151*         22 KURT(JOIN+I)=0.                                                              000560
00266     152*      C                                                                                  000560
00266     153*      C   COVARIANCE MUST BE POSITIVE DEFINITE                                      CH    000560
00270     154*            IF(VOLIN(JOIN).LE.0.) PRINT 653,LS,JOIN,VOLIN(LS),VOLIN(JOIN)                000561
00271     155*        653 FORMAT(' VOLUME ERROR IN JOIN: CLASSES,VOLUMES',2I5,2E10.5)                   000577
00300     156*            VOLIN(JOIN)=ABS(VOLIN(JOIN))*.875651076 3E-26*(6.283185307/W(JOIN))          000577
00300     157*           1       **MQ                                                                  000577
00300     158*      C                                                                            CH    000577
```

```
00301     159*      C   PUT VOLUME (VOLIN) IN INTERNAL FORM, CALCULATE VOLRT, INIT DCON                    000577
00301     160*          VOLRT(JOIN)=SQRT(VOLIN(JOIN))                                                     000613
00302     161*          DCON(JOIN)=0.DCON                                                                 000621
00303     162*          OW(JOIN)=W(JOIN)                                                                  000623
00303     163*      C                                                                                     000623
00303     164*      C                                                                                     000623
00303     165*      C                                                                                     000623
00303     166*      C                        *** PRINT DATA FOR NEW CLUSTER ***                           000623
00303     167*      C                                                                                     000623
00303     168*      C   PRINT OUT (IF DESIRED)                                                            000623
00304     169*          CALL CLPR(JOIN,-1,SUM,SKEW,KURT)                                                  000625
00305     170*          CALL CLPR(KA,-2,SUM,SKEW,KURT)                                                    000634
00306     171*          CALL CLPR(KB,-2,SUM,SKEW,KURT)                                                    000643
00307     172*          RETURN                                                                            000652
00310     173*          END                                                                               000763

      END OF COMPILATION:       NO  DIAGNOSTICS.
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S LAREAD,LAREAD,LAREAD
FOR S0E3-04/18/78-01:31:51 (0,0)

   FUNCTION LAREAD       ENTRY POINT 000522

   STORAGE USED: CODE(1) 000557; DATA(0) 000225; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003    NXTCHR
    0004    CMERR
    0005    NRDU$
    0006    NI02$
    0007    NI03$
    0010    NWDU$
    0011    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000013 1L        0001   000470 10L       0001   000063 11L        0001   000004 120G      0001   000005 123G
    0000   000154 13F       0001   000400 14L       0000   000152 15F        0001   000314 2L        0000   000151 20F
    0001   000041 21L       0001   000050 22L       0000   000152 23F        0001   000031 24L       0001   000345 263G
    0001   000270 3L        0001   000455 30L       0001   000416 306G       0001   000463 31L       0001   000446 35L
    0001   000104 4L        0001   000252 6L        0001   000230 7L         0001   000244 8L        0000 I 000127 AST
    0000 I 000123 BLANK     0000 I 000131 C         0000 I 000001 CARD       0000 I 000150 CNT       0000 I 000135 COL
    0000 I 000124 COMMA     0000 I 000126 CP        0000 I 000130 ENDBCD     0000 I 000133 I         0000 I 000136 II
    0000   000204 INJP$     0000 I 000134 J         0000 I 000137 K          0000 I 000140 KK        0000 I 000000 LAREAD
    0000 I 000147 N         0000 I 000143 NT1       0000 I 000145 NT2        0000 I 000144 NT3       0000 I 000146 NT4
    0000 I 000141 NUM       0000 I 000142 NW        0003 I 000000 NXTCHR     0000 I 000125 OP        0000 I 000132 S
    0000 I 000077 VER

 00100      C        //LAREAD                                                     LARE0000    000005
 00101               FUNCTION LAREAD(FLDNAM,VERTCS,FLDINF,NC)                                 000005
 00103               IMPLICIT INTEGER(A-Z)                                                    000005
 00104               DIMENSION CARD(62),FLDINF(6),VERTCS(2,11),VER(2,10)                      000005
 00105               DATA BLANK/' '/,COMMA/','/,OP/'('/,CP/')'/,                              000005
 00105              * AST/'*'/,ENDBCD/'$END*'/                                                000005
 00114               DATA C/'CLASSN'/,S/'SUBCLA'/                                             000005
 00117               DO 50 I=1,2                                                              000005
 00122               DO 50 J=1,10                                                             000005
 00125          50   VER(I,J)=0                                                               000005
 00125      C READ IN FIRST CARD                                                              000005
 00130            1  READ(5,20) FLDNAM                                                        000005
 00133           20  FORMAT(A6)                                                               000013
 00134               IF(FLDNAM.NE.C) GO TO 24                                                 000020
 00136               LAREAD=-1                                                                000020
 00137               RETURN                                                                   000023
 00140           24  IF(FLDNAM.NE.S) GO TO 21                                                 000025
 00142               LAREAD=-2                                                                000031
 00143               RETURN                                                                   000033
                                                                                              000035
```

```
00144 20*        21 IF(FLDNAM.NE.ENDBCD) GO TO 22                                  000042
00146 21*           LAREAD=0                                                       000043
00147 22*           RETURN                                                         000044
00147 23*     C REREAD FIRST CARD                                                  000044
00150 24*        22 READ(30,23) CARD                                               000050
00153 25*        23 FORMAT(10X,62A1)                                               000057
00156 26*           COL=0                                                          000057
00159 27*           II=0                                                           000060
00156 28*           NC=0                                                           000061
00156 29*     C FIND (                                                             000061
00157 30*        11 J=NXTCHR(CARD,COL)                                             000063
00160 31*           IF(J.EQ.BLANK) GO TO 1                                         000067
00162 32*           IF(J.EQ.AST) GO TO 10                                          000071
00165 33*           IF(J.NE.OP) GO TO 3                                            000074
00166 34*         9 I=0                                                            000077
00167 35*           K=0                                                            000100
00170 36*           KK=0                                                           000101
00171 37*           NUM=0                                                          000102
00172 38*         4 COL=COL+1                                                      000104
00173 39*           IF(COL .GT. 62) GO TO 35                                       000106
00175 40*           IF(CARD(COL).EQ.BLANK) GO TO 4                                 000111
00177 41*           IF((I.EQ.0).AND.(CARD(COL).EQ.COMMA)) GO TO 3                  000115
00201 42*           IF((I.EQ.1).AND.(CARD(COL).EQ.COMMA)) GO TO 7                  000127
00203 43*           IF((KK.EQ.0).AND.(CARD(COL).EQ.CP)) GO TO 3                    000143
00205 44*           IF((KK.EQ.1).AND.(CARD(COL).EQ.CP)) GO TO 8                    000155
00207 45*           NW=FLD(ABS(0),6,CARD(COL))                                     000171
00210 46*           NW=NW-48                                                       000173
00211 47*           NUM=10*NUM+NW                                                  000176
00212 48*           IF((NW.LT.0).OR.(NW.GT.9)) GO TO 3                             000202
00214 49*           I=1                                                            000217
00215 50*           IF(K.EQ.1) KK=1                                                000221
00217 51*           GO TO 4                                                        000226
00220 52*         7 IF(II.EQ.0) GO TO 30                                           000230
00220 53*     C VERTEX SAMPLE NUMBER                                               000230
00222 54*           NC=NC+1                                                        000231
00223 55*           VER(1,NC)=NUM                                                  000234
00224 56*           K=1                                                            000237
00225 57*           NUM=0                                                          000241
00226 58*           GO TO 4                                                        000242
00227 59*         8 IF(II.EQ.0) GO TO 31                                           000244
00227 60*     C VERTEX LINE NUMBER                                                 000244
00231 61*           VER(2,NC)=NUM                                                  000245
00231 62*     C CHECK FOR COMMA OR ASTERISK                                        000245
00232 63*         6 J=NXTCHR(CARD,COL)                                             000252
00233 64*           IF(J.EQ.BLANK) GO TO 2                                         000256
00235 65*           IF(J.EQ.AST) GO TO 10                                          000260
00237 66*           IF(J.NE.COMMA) GO TO 3                                         000263
00241 67*           GO TO 11                                                       000266
00242 68*         3 WRITE(6,13) CARD                                               000270
00245 69*        13 FORMAT(' ERROR IN FIELD CARD TERMINATING RUN'/10X,62A1)         000277
00246 70*           CALL CMERR                                                     000277
00247 71*         5 WRITE(6,15) CARD                                               000301
00252 72*        15 FORMAT(10X,62A1/' INCORRECT FIELD CARD,TERMINATING RUN')       000311
00253 73*           CALL CMERR                                                     000311
00253 74*     C DETERMINE RECTANGULAR FIELD COORDINATES                            000311
00254 75*         2 IF((NC.LT.1).OR.(NC.GT.10)) GO TO 3                            000314
00256 76*           NT1=0                                                          000331
```

```
00257          NT3=0                                              000334
00260          NT2=1000000                                        000335
00261          NT4=1000000                                        000337
00262          DO 14 N=1,NC                                       000345
00265          IF(VER(1,N).GT.NT1) NT1=VER(1,N)                   000345
00267          IF(VER(2,N).LT.NT4) NT4=VER(2,N)                   000353
00271          IF(VER(2,N).GT.NT3) NT3=VER(2,N)                   000361
00273          IF(VER(1,N).GE.NT2) GO TO 14                       000367
00275          NT2=VER(1,N)                                       000373
00276          CNT=N                                              000375
00277       14 CONTINUE                                           000402
00301          FLDINF(1)=NT4                                      000402
00302          FLDINF(2)=NT3                                      000404
00303          FLDINF(4)=NT2                                      000406
00304          FLDINF(5)=NT1                                      000410
00304    C SET UP VERTICES IN CLOCKWISE ORDER WITH SMALLEST SAMPLE FIRST  000410
00305          DO 32 I=1,NC                                       000416
00310          IF(CNT.GT.NC) CNT=1                                000416
00312          VERTCS(1,I)=VER(1,CNT)                             000423
00313          VERTCS(2,I)=VER(2,CNT)                             000427
00314       32 CNT=CNT+1                                          000431
00316          VERTCS(1,NC+1)=VERTCS(1,1)                         000435
00317          VERTCS(2,NC+1)=VERTCS(2,1)                         000443
00320       35 LAREAD=1                                           000446
00321          NC=NC+1                                            000447
00322          RETURN                                             000451
00323       30 FLDINF(6)=NUM                                      000455
00324          K=1                                                000456
00325          NUM=0                                              000460
00326          GO TO 4                                            000461
00327       31 FLDINF(3)=NUM                                      000463
00330          II=1                                               000464
00331          GO TO 6                                            000466
00331    C READ CONTINUATION CARD                                 000466
00332       10 READ(5,23) CARD                                    000470
00335          COL=0                                              000477
00336          GO TO 11                                           000500
00337          END                                                000556
```

END OF COMPILATION: NO DIAGNOSTICS.

```
@FOR,S MINV,MINV,MINV
FOR SDE3-04/18/78-01:32:11 (0,0)

   SUBROUTINE MINV      ENTRY POINT 000303

   STORAGE USED: CODE(1) 000333; DATA(0) 000055; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003   CLUS   002017
    [004   MISC   000113
    [005   STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006   NERR3$

   STORAGE ASSIGNMENT. (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001    000065 134G       0001    000066 137G       0001    000121 147G       0001    000155 156G       0001    000167 164G
    [001    000227 172G       0001    000240 22L        0003 R  000020 ALINK      0004    000064 AMM        0004    000075 AMOFAC
    [004    000077 AMOMAX     0004    000076 AMOMIN     0004    000100 AMORAT     0004    000054 AMQ        0004    000107 BETTER
    [004    000102 BIAS       0003 R  000027 CIN        0005    000001 CONLV      0004    000111 CORLEN     0003 R  000030 CTOT
    [003 R  000041 DCON       0004    000053 DELT       0003 R  000043 DISS       0004    000073 DWFAC      0004    000061 ELIMTH
    [004    000052 EPS        0003 R  000050 GEN        0004    000074 GRACTM     0003 R  000047 GREF       0000 I  000001 I
    [003 I  000023 IDADJ      0003 I  000024 INDEX      0004    000066 INDXVL     0000    000005 INJPS      0000 I  000003 IP
    [000 I  000002 J          0003 I  000000 JUNK       0004    000051 KROOT      0003 I  000020 LINK       0004    000003 LKURT
    [004    000007 LOSUM      0004    000004 LOVAR      0004 I  000002 LR         0004    000006 LSKEW      0003 I  000021 LSUBS
    [004    000005 LSUM       0003 I  000022 LSUPER     0004 I  000005 LV         0004    000002 LVRIN      0005 R  000012 MACCEL
    [004    000001 MM         0004    000110 MODE       0004 I  000000 MQ         0004 I  000011 MXAR       0003    000014 NARL
    [004    000010 NINCLS     0004    000070 NPTS0      0003 I  000043 NSYSB      0003 I  000020 NTB        0003    000016 NTBSZM
    [003    000015 NTOP       0003    000017 NWANT      0003 R  000045 OCIN       0004    000055 ODCON      0003 R  000047 ODEN
    [003 R  000046 OPRIOR     0003 R  000035 OPROP      0003 R  000036 OW         0005 R  000010 PACCEL     0003 R  000050 PCOND
    [003 R  000025 PCUM       0004    000103 PJOIN      0003 R  000043 PPASS      0003 R  000042 PORAT      0004    000071 PORATH
    [003 R  000026 PRIRCM     0003 R  000031 PROP       0003 R  000044 PST        0004    000065 SBLTH      0004    000062 SEPTH
    [005    000002 SKBND      0005    000003 SKCHI      0004    000112 SPCOR      0003 R  000032 SPFAC      0004    000072 SPMVTH
    [005    000004 TRBND      0005    000005 TRCHI      0005    000006 URKBND     0005    000007 URKCHI     0005 R  000014 VACCEL
    [004    000063 VFAC       0003 R  000037 VOLIN      0004    000101 VOLLIM     0003 R  000040 VOLRT      0003 R  000050 VRIN
    [004    000104 VRJOIN     0003 R  000034 W          0003 R  000033 WADJ       0004    000060 WADJIN     0005    000000 WAIT
    [004    000106 WDELSM     0004    000067 WFAC       0004    000105 WSIM       0004    000050 WTINIT     0004    000056 XOVFLO
    [004    000057 XUNFLO     0000 R  000000 Z

000001  *DIAGNOSTIC* THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*            SUBROUTINE MINV(A,B,C,VOL)                                          000041
00101       2*       C  THIS ROUTINE CALCULATES A=THE INVERSE OF C, A=C**-1.  IT ALSO          000041
00101       3*       C     RETURNS THE DETERMINANT OF C IN VOL.  THE SQUARE ARRAY            000041
00101       4*       C     B IS TEMPORARY STORAGE, AND MAY BE IDENTICAL TO C.                000041
00101       5*       C     VOL=-ABS(DET(C)) IF C IS NOT POSITIVE DEFINITE.                   000041
00101       6*            INCLUDE MISH                                                      000041
00131       7*            REAL Z, A(MQ,MQ),B(MQ,MQ),C(MQ,MQ)                                000041
```

```
00132            VOL=1.                                   000041
00133            DO 11 I=1,MQ                             000066
00136            DO 10 J=1,MQ                             000066
00141            B(I,J)=C(I,J)                            000066
00142         10 A(I,J)=0.                                000067
00144         11 A(I,I)=1.                                000071
00146            DO 22 I=1,MQ                             000121
00151            VOL=VOL*B(I,I)                           000140
00152            IF(B(I,I).LE.0.) VOL=-ABS(VOL)           000143
00154            Z=1./B(I,I)                              000153
00155            DO 21 J=1,MQ                             000155
00160            B(I,J)=B(I,J)*Z                          000155
00161         21 A(I,J)=A(I,J)*Z                          000157
00163            DO 22 IP=1,MQ                            000167
00166            IF(IP.EQ.I) GO TO 22                     000200
00170            Z=B(IP,I)                                000203
00171            DO 23 J=1,MQ                             000227
00174            B(IP,J)=B(IP,J)-B(I,J)*Z                 000227
00175         23 A(IP,J)=A(IP,J)-A(I,J)*Z                 000232
00177         22 CONTINUE                                 000253
00202            RETURN                                   000253
00203            END                                      000332
```

END OF COMPILATION: 1 DIAGNOSTICS.

```
@FOR,S MLT,MLT,MLT
FOR (CE3=04/18/78-01:32:34 (0,0)

   SUBROUTINE MLT       ENTRY POINT 00010*

   STORAGE USED: CODE(1) 000117; DATA(0) 000033; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003    CLUS    002017
    [004    MISC    000113
    [005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001    000040 133G      0001    000043 136S      0001    000045 142G      0003 R 000020 ALINK     0004    000064 AMM
    [004    000075 AMOFAC    0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ
    [004    000107 BETTER    0004    000102 BIAS      0003 R 000027 CIN       0005    000001 CONLV     0004    000111 CORLEN
    [003 P  000030 CTOT      0003 R  000041 DCON      0004    000053 DELT      0003 R  000043 DISS      0004    000073 DWFAC
    [004    000061 ELIMTH    0004    000052 EPS       0003 R 000050 GEN       0004    000074 GRACTM    0003 R  000047 GREF
    [000 I  000001 I         0003 I  000023 IDADJ     0003 I 000024 INDEX     0004    000066 INDXVL    0000    000005 INJPS
    [000 I  000002 J         0003 I  000000 JUNK      0000 I 000003 K         0004    000051 KROOT     0003 I  000020 LINK
    [004    000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR     0004 I 000002 LR        0004    000006 LSKEW
    [003 I  000021 LSUBS     0004    000005 LSUM      0003 I 000022 LSUPER    0004 I 000005 LV        0004    000002 LVRIN
    [005 P  000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I 000000 MQ        0004 I  000011 MXAR
    [003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSQ     0003 I 000043 NSYMB     0003 I  000020 NTB
    [003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT     0003 R 000045 OCIN      0004    000055 ODCON
    [003 P  000047 ODEN      0003 R  000046 OPRIOR    0003 R 000035 OPROP     0003 R 000036 OW        0005 R  000010 PACCEL
    [003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R 000043 PPASS     0003 R  000042 PQRAT
    [004    000071 PQRATH    0003 P  000026 PRIRCM    0003 R 000031 PROP      0003 R 000044 PST       0004    000065 SBLTH
    [004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR     0003 R  000032 SPFAC
    [004    000072 SPMVTH    0000 R  000000 SUM       0005    000004 TPBND     0005    000005 TPCHI     0005    000006 URKBND
    [005    000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R 000037 VCLIN     0004    000101 VOLLIM
    [003 R  000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R 000034 W         0003 R  000033 WADJ
    [004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIM
    [004    000050 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

00000[  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00010[      1*          SUBROUTINE MLT(A,B,C)                                                    000043
00010[      2*              INCLUDE *ISH                                                         000043
00013[      3*              REAL SUM,A(MQ,MQ),B(MQ,MQ),C(MQ,MQ)                                 000043
00013[      4*              DO 13 I=1,MQ                                                         000043
00013[      5*              DO 13 J=1,MQ                                                         000043
00014[      6*              SUM=0.                                                               000043
00014[      7*              DO 12 K=1,MQ                                                         000045
00014[      8*           12 SUM=SUM+B(I,K)*C(K,J)                                                000045
```

FINAL PAGE IS OF POOR QUALITY

```
00146      9*       13 A(I,J)=SUM
00151     10*          RETURN                                    000051
00152     11*          END                                       000063
                                                                 000116

      END OF COMPILATION:       1 DIAGNOSTICS.
```

```
@FOR,S MPVS,MPVS,MPVS
FOR SCE3-04/18/78-01:32:37 (0,0)

   SUBROUTINE MPVS      ENTRY POINT 000054

   STORAGE USED: CODE(1) 000078; DATA(0) 000024; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   CLUS    002017
    0004   MISC    000113
    0005   STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000012 1346       0001    000021 1376       0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    0004    000077 AMOMAX     0004    000076 AMOMIN     0004    000100 APORAT   0004    000054 AMQ       0004    000107 BETTER
    0004    000102 BIAS       0003 R 000027 CIN       0005    000C01 CONLV    0004    000111 CORLEN    0003 R 000030 CTOT
    0003 P 000041 DCON        0004    000053 DELT       0003 R 000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH
    0004    000052 EPS        0003 R 000050 GEN       0004    000074 GRACTM   0003 R 000047 GREF      0000 I 000001 I
    0003 I 000023 IDADJ       0003 I 000024 INDEX     0004    000066 INDXVL   0000    000004 INJP$     0000 I 000002 J
    0003 I 000000 JUNK        0004    000051 KROOT      0003 I 000020 LINK      0004    000003 LKURT     0000 I 000000 LOC
    0004    000007 LOSUM      0004    000004 LOVAR      0004 I 000002 LR        0004    000006 LSKEW     0003 I 000021 LSUBS
    0004    000005 LSUM       0003 I 000022 LSUPER    0004 I 000005 LV        0004    000002 LVRIN     0005 R 000012 MACCEL
    0004    000001 MM         0004    000110 MODE       0004 I 000000 MQ        0004 I 000011 MXAR      0003    000014 NARL
    0004    000010 NINCLS     0004    000070 NPTSO      0003 I 000043 NSYMB     0003 I 000020 NTB       0003    000016 NTBSZM
    0003    000015 NTOP       0003    000017 NWANT      0003 R 000045 OCIN      0004    000055 ODCON     0003 R 000047 ODEN
    0003 R 000046 OPRIOR      0003 R 000035 OPROP       0003 R 000036 OW        0005 R 000010 PACCEL    0003 R 000050 PCOND
    0003 R 000025 PCUM        0004    000103 PJOIN      0003 R 000043 PPASS     0003 R 000042 PQRAT     0004    000071 PQRATH
    0003 R 000026 PRIPCM      0003 R 000031 PROP        0003 R 000044 PST       0004    000065 SBLTH     0004    000062 SEPTH
    0005    000002 SKBND      0005    000003 SKCHI      0004    000112 SPCOR    0003 R 000032 SPFAC     0004    000072 SPMVTH
    0005    000004 TRBND      0005    000005 TRCHI      0005    000006 URKBND   0005    000007 URKCHI    0005 R 000014 VACCEL
    0004    000063 VFAC       0003 R 000037 VOLIN       0004    000101 VOLLIM   0003 R 000040 VOLRT     0003 R 000050 VRIN
    0004    000104 VRJOIN     0003 R 000034 W         0003 R 000033 WADJ      0004    000060 WADJIN    0005    000000 WAIT
    0004    000106 WDELSM     0004    000067 WFAC       0004    000105 WSIM     0004    000050 WTINIT    0004    000056 XOVFLO
    0004    000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*          SUBROUTINE MPVS(AM,C,V)                                                    000005
00101      2*      C     SETS AM=AM+V*V*C    (TENSOR PRODUCT)                                     000005
00103      3*          INCLUDE MISH                                                               000005
00131      4*          REAL AM(475),V(30)                                                         000005
00132      5*          LOC=0                                                                      000005
00133      6*          DO 10 I=1,MO                                                               000012
00136      7*          DO 10 J=1,I                                                                000021
00141      8*          LOC=LOC+1                                                                  000021
```

```
00142      9*          10 AM(LOC)=AM(LOC)+V(I)*V(J)*C                    000025
00145     10*             RETURN                                         000036
00146     11*             END                                            000067

 END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S MTVEC,MTVEC,MTVEC
FOR S0E3-04/18/78-01:32:39 (0,0)

   SUBROUTINE MTVEC      ENTRY POINT 000046

   STORAGE USED: CODE(1) 000061; DATA(0) 000021; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003     CLUS     002017
    0004     MISC     000113
    0005     STPAR    000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006     NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000017 1336      0001     000023 1376      0003 R 000020 ALINK     0004     000064 AMM       0004     000075 AMOFAC
    0004     000077 AMOMAX    0004     000076 AMOMIN    0004     000100 AMORAT    0004     000054 AMQ       0004     000107 BETTER
    0004     000102 BIAS      0003 R 000027 CIN       0005     000001 CONLV     0004     000111 CORLEN    0003 R 000030 CTOT
    0003 R 000041 DCON      0004     000053 DELT      0003 R 000043 DISS      0004     000073 DWFAC     0004     000061 ELIMTH
    0004     000052 EPS       0003 R 000050 GEN       0004     000074 GRACTM    0003 R 000047 GREF      0000 I 000001 I
    0003 I 000023 IGADJ     0003 I 000024 INDEX     0004     000066 INDXVL    0000     000004 INJP$     0000 I 000002 J
    0003 I 000000 JUNK      0004     000051 KROOT     0003 I 000020 LINK      0004     000003 LKURT     0004     000007 LOSUM
    0004     000004 LOVAR     0004 I 000002 LR        0004     000006 LSKEW     0003 I 000021 LSUBS     0004     000005 LSUM
    0003 I 000022 LSUPER    0004 I 000005 LV        0004     000002 LVRIN     0005 R 000012 MACCEL    0004     000001 MM
    0004     000110 MODE      0004 I 000000 MQ        0004 I 000011 MXAR      0003     000014 NARL      0004     000010 NINCLS
    0004     000070 NPTSG     0003 I 000043 NSYMB     0003 I 000020 NTB       0003     000016 NTBSZM    0003     000015 NTOP
    0003     000017 NWANT     0003 R 000045 OCIN      0004     000055 ODCON     0003 R 000047 ODEN      0003 R 000046 OPRIOR
    0003 P 000035 OPROP     0003 R 000036 OW        0005 R 000010 PACCEL    0003 R 000050 PCOND     0003 R 000025 PCUM
    0004     000103 PJOIN     0003 R 000043 PPASS     0003 R 000042 PORAT     0004     000071 PORATH    0003 R 000026 PRIRCM
    0003 P 000031 PROP      0003 R 000044 PST       0004     000055 SBLTH     0004     000062 SEPTH     0005     000002 SKBND
    0005     000003 SKCHI     0004     000112 SPCOR     0003 R 000032 SPFAC     0004     000072 SPMVTH    0000 R 000000 SUM
    0005     000004 TRBND     0005     000005 TRCHI     0005     000006 URKBND    0005     000007 URKCHI    0005 R 000014 VACCEL
    0004     000063 VFAC      0003 R 000037 VOLIN     0004     000101 VOLLIM    0003 R 000040 VOLRT     0003 R 000050 VRIN
    0004     000104 VRJOIN    0003 R 000034 W         0003 R 000033 WADJ      0004     000060 WADJIN    0005     000000 WAIT
    0004     000106 WDELSM    0004     000067 WFAC      0004     000105 WSIM      0004     000050 WTINIT    0004     000056 XOVFLO
    0004     000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*             SUBROUTINE MTVEC(U,A,V)                                              000017
00103      2*             INCLUDE MISH                                                         000017
00131      3*             REAL SUM,U(MQ),V(MQ),A(MQ,MQ)                                        000017
00132      4*             DO 13 I=1,MQ                                                         000017
00135      5*             SUM=0.                                                               000017
00136      6*             DO 12 J=1,MQ                                                         000023
00141      7*          12 SUM=SUM+A(J,I)*V(J)                                                  000023
00143      8*          13 U(I)=SUM                                                             000027
```

```
0014E      9*          RETURN                                          000031
0014E     10*          END                                             000060

END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S MULTI,MULTI,MULTI
FOR SOE3-04/18/78-01:32:42 (0,0)

   SUBROUTINE MULTI     ENTRY POINT 000074

   STORAGE USED: CODE(1) 000105; DATA(0) 000011; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    [003    INFORM 000407
    [004    CLUSTR 000131
    [005    CLUS   002017
    [006    MISC   000113
    [007    STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [010    DATFIX
    [011    ALFREE
    [012    CLINIT
    [013    STATIS
    [014    CLDUMP
    [015    NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [005 R 000020 ALINK      0006    000064 AMM      0006    000075 AMOFAC   0006    000077 AMOMAX   0006    000076 AMONIN
    [006   000100 AMORAT     0006    000054 AMQ      0004    000123 AREA     0003    000351 BESTVC   0006    000107 BETTER
    [006   000102 BIAS       0004 I  000115 BUFPIX   0004 I  000116 BUFTOT   0005 R  000027 CIN      0004 I  000002 CLSNAM
    [007   000001 CONLV      0006    000111 CORLEN   0005 R  000030 CTOT     0003    000053 DATAPE   0005 R  000041 DCON
    [006   000053 DELT       0005 R  000043 DISS     0003    000112 DIVSIZ   0004    000113 DUMP     0003    000062 DUPSYM
    [006   000073 DWFAC      0006    000061 ELIMTH   0006    000052 EPS      0003    000217 FETVC2   0003    000161 FETVEC
    [003   000071 FILESV     0005 R  000050 GEN      0006    000074 GRACTM   0005 R  000047 GREF     0003    000000 HEAD
    [003   000255 HISVEC     0004    000000 IBEGIN   0003    000123 IBLOCK   0005 I  000023 IDADJ    0005 I  000024 INDEX
    [006   000066 INDXVL     0000    000000 INJPS    0003    000313 INVERT   0004    000003 IPT      0004    000130 IQ1
    [005 I 000000 JUNK       0003    000113 KEEPLV   0004 I  000107 KLBC     0006 I  000051 KROOT    0004    000127 LBUF
    [004   000121 LBUFD      0005 I  000020 LINK     0006 I  000003 LKURT    0004    000102 LNCAT    0006 I  000007 LOSUM
    [006 I 000004 LOVAR      0006 I  000002 LR       0006 I  000006 LSKEW    0005 I  000021 LSUBS    0006 I  000005 LSUM
    [005 I 000022 LSUPER     0006 I  000005 LV       0006    000002 LVRIN    0007 R  000012 MACCEL   0003    000052 MAPTAP
    [004   000122 MAXBF      0003    000072 MAXCLS   0003    000064 MAXDIV   0003    000055 MAXFET   0003    000074 MAXFLD
    [003   000065 MINDIV     0006 I  000001 MM       0006    000110 MODE     0006 I  000000 MQ       0006 I  000011 MXAR
    [005   000014 NARL       0004    000117 NBUFSD   0004    000120 NDUMP    0006 I  000010 NINCLS    0004    000106 NIT
    [003   000073 NOCLS2     0003    000121 NOCLS3   0003 I  000100 NOFEAT   0003    000101 NOFET2   0003    000102 NOFET4
    [004   000004 NOFLD      0003    000075 NOFLO2   0003    000076 NOFLO3   0003    000111 NOGRP    0003    000110 NOHIST
    [003   000107 NOSPEC     0003    000077 NOTRFD   0004    000126 NPTS     0006    000070 NPTSO    0005 I  000043 NSYMB
    [005 I 000020 NTB        0005    000016 NTBSZM   0005    000015 NTOP     0005    000017 NWANT    0004    000125 NWDRS
    [004   000124 NWDS       0005 R  000045 OCIN     0006    000055 ODCON    0005 R  000047 ODEN     0005 R  000046 OPRIOR
    [005 R 000035 OPROP      0005 R  000036 OW       0007 R  000010 PACCEL   0003    000056 PAGSIZ   0005 R  000050 PCOND
    [003   000122 PCISZ      0005 R  000025 PCUM     0006    000103 PJOIN    0005 R  000043 PPASS    0005 R  000042 PORAT
    [006   000071 PORATH     0005 R  000026 PRIRCM   0004 I  000103 PRNT     0005 R  000031 PROP     0004 I  000111 PROUT
    [003   000114 PRTLEV     0004 I  000110 PRIME    0005 R  000044 PST      0003    000054 SAVTAP   0006    000065 SBLTH
    [004 I 000114 SCRAM1     0006    000062 SEPTH    0003    000067 SERIAL   0007    000002 SKBND    0007    000003 SKCHI
    [003   000120 SPCBAS     0006    000112 SPCOR    0005 R  000032 SPFAC    0003    000066 SPLMAX   0006    000072 SPMVTH
```

```
C004  I 000005 SYM        0003   000057 TAPCHK     0003   000070 TAPESV     0003   000053 THRSYM     0004  I 000112 TOTPIX
C004  I 000001 TOTWRD     0007   000004 TRBND      0007   000005 TRCHI      0003   000060 TRNSYM     0003    000061 TSTSYM
C007    000006 URKBND     0007   000007 URKCHI     0007 R 000014 VACCEL     0003   000103 VARSIZ     0003    000104 VARSZ2
C003    000105 VARSZ4     0006   000063 VFAC       0005 R 000037 VOLIN      0006   000101 VOLLIM     0005 R  000040 VOLRT
C005  P 000050 VRIN       0006   000104 VRJOIN     0005 R 000034 W          0005 R 000033 WADJ       0006    000060 WADJIN
C007    000000 WAIT       0006   000106 WDELSM     0006   000067 WFAC       0006   000105 WSIM       0006    000050 WTINIT
C003    000116 XHGH       0003   000117 XLOW       0006   000056 XOVFLO     0003   000106 XSIZ       0006    000057 XUNFLO
C003    000115 YSIZ
```

```
0000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
0010(      1*      C  MULTI                                                                          000000
0010(      2*           SUBROUTINE MULTI(PV)                                                         000000
0010(      3*      C                                                                                 000000
0010(      4*      C                                                                                 000000
0010(      5*      C  PURPOSE--                                                                      000000
0010(      6*      C                                                                                 000000
0010(      7*      C                                                                                 000000
0010(      8*      C                                                                                 000000
0010(      9*      C                                                                                 000000
0010(     10*           INCLUDE COMBK1                                                               000000
0010(     11*           INCLUDE CMBK1D                                                               000000
0011(     12*           INCLUDE MISH                                                                 000000
0011(     13*      C                                                                                 000000
0011(     14*      C                                                                                 000000
0014(     15*           DIMENSION   PV(I,1)                                                          000000
0014(     16*      C                                                                          CH     000000
0014(     17*      C  CALL DATFIX TO INITIALIZE VARIABLES                                            000000
0014(     18*           CALL DATFIX                                                                  000000
0014(     19*           MQ=NOFEAT                                                                    000001
0014(     20*           MM=(MQ*(MQ+1))/2                                                             000003
0014(     21*      C                                                                                 000003
0014(     22*      C  CALL ALFREE TO FREE STRING STARTED BY KLBC                                     000003
0014(     23*           CALL ALFREE(KLBC,NINCLS)                                                     000007
0014(     24*      C                                                                          CH     000007
0014(     25*      C  CALL CLINIT TO INITIALIZE THE CLUSTERING ALGORITHM                             000007
0014(     26*           CALL CLINIT(KLBC)                                                            000013
0015(     27*           KROOT=KLBC                                                                   000023
0015(     28*      C                                                                                 000023
0015(     29*      C  CALL STATIS TO CLASSIFY EACH POINT AND UPDATE STATISTICS                       000023
0015(     30*           CALL STATIS(XLBC,PV,GEN(LSUM),GEN(LSKEW),GEN(LKURT),GEN(LOSUM),              000025
0015(     31*          *  GEN(LOVAR))                                                               000025
0015(     32*      C K: PRINT OUT ALL CLASSES.                                                       000025
0015(     33*           PROUT=3                                                                      000050
0015(     34*      C                                                                                 000050
0015(     35*      C  CALL CLDUMP TO PRINT ALL CLASSES FOR ENTIRE TREE UNDER KLBC                    000050
0015(     36*           CALL CLDUMP(KLBC)                                                            000052
0015(     37*           PROUT=1                                                                      000055
0015(     38*           RETURN                                                                       000057
0015(     39*           END                                                                          000104

      END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S MVEC,MVEC,MVEC
FOR SOE3-04/18/78-01:32:53 (0,0)

   SUBROUTINE MVEC      ENTRY POINT 000060

   STORAGE USED: CODE(1) 000074; DATA(0) 000024; BLANK COMMON(2) 0000

    COMMON BLOCKS:

    [003    CLUS    002017
    [004    MISC    000113
    [005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    [006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    [001     000025 1336       0001    000031 1376      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    [004     000077 AMOMAX     0004    000076 AMOMIN    0004   000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER
    [004     000102 BIAS       0003 R  000027 CIN       0005   000001 CONLV     0004    000111 CORLEN    0003 R  000036 CTOT
    [003  P  000041 DCON       0004    000053 DELT      0003 R 000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH
    [004     000052 EPS        0003 R  000050 GEN       0004   000074 GRACTM    0003 R  000047 GPEF      0000 I  000001 I
    [003  I  000023 IDADJ      0003 I  000024 INDEX     0004   000066 INDXVL    0000    000004 INJPS     0000 I  000002 J
    [003  I  000000 JUNK       0004    000051 KROOT     0003 I 000020 LINK      0004    000003 LKURT     0004    000007 LOSUM
    [004     000004 LOVAR      0004 I  000002 LR        0004   000096 LSKEW     0003 I  000021 LSUBS     0004    000005 LSUM
    [003  I  000022 LSUPER     0004 I  000005 LV        0004   000002 LVRIN     0005 R  000012 MACCEL    0004    000001 MM
    [004     000110 MODE       0004 I  000000 MQ        0004 I 000011 MXAR      0003    000014 NAR1      0004    000010 NINCLS
    [004     000070 NPTSO      0003 I  000043 NSYMB     0003 I 000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP
    [003     000017 NWANT      0003 R  000045 OCIN      0004   000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR
    [003  R  000035 OPROP      0003 R  000036 OW        0005 R 000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM
    [004     000103 PJOIN      0003 R  000043 PPASS     0003 R 000042 PORAT     0004    000071 PORATH    0003 R  000026 PRIRCM
    [003  P  000031 PROP       0003 R  000044 PST       0004   000065 SELTH     0004    000062 SEPTH     0005    000002 SKBND
    [005     000003 SKCHI      0004    000112 SPCOR     0003 R 000032 SPFAC     0004    000072 SPMVTH    0000 R  000000 SUM
    [005     000004 TRBND      0005    000005 TRCHI     0005   000006 URKBND    0005    000007 URKCHI    0005 R  000014 VACCEL
    [004     000063 VFAC       0003 R  000037 VOLIN     0004   000101 VOLLIM    0003 R  000040 VOLRT     0003 R  000050 VRIN
    [004     000104 VRJOIN     0003 R  000034 W         0003 R 000033 WADJ      0004    000060 WADJIN    0005    000000 WAIT
    [004     000106 WDELSM     0004    000067 WFAC      0004   000105 WSIM      0004    000050 WTINIT    0004    000056 XOVFLO
    [004     000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00010 1      1*          SUBROUTINE MVEC(U,A,V)                                                    000025
00010 1      2*          INCLUDE MISH                                                              000025
00013 1      3*          REAL SUM,U(MQ),V(MQ),A(MQ,MQ)                                             000025
00013 2      4*          DO 13 I=1,MQ                                                              000025
00013 5      5*          SUM=0.                                                                    000025
00013 6      6*          DO 12 J=1,MQ                                                              000031
00014 1      7*       12 SUM=SUM+A(I,J)*V(J)                                                       000031
00014 3      8*       13 U(I)=SUM                                                                  000035
```

```
00145      9*        RETURN                                                   000042
00146     10*        END                                                      000073

      END OF COMPILATION:        1 DIAGNOSTICS.
```

```
@FOR,S READTP,READTP,READTP
FOR SOE3-04/18/78-01:32:56 (1,0)

   SUBROUTINE READTP     ENTRY POINT 000447

   STORAGE USED: CODE(1) 001652; DATA(0) 000427; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    INFORM 000407
    C004    CLUSTR 000131
    C005    TEST   000231

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    RINIT
    C007    TAPHDR
    C010    LAREAD
    C011    FLDINT
    C012    LINERD
    C013    FDLINT
    C014    CMERR
    C015    RWRITE
    C016    ZOR
    C017    RREAD
    C020    NWDU$
    C021    NIO2$
    C022    NIO1$
    C023    NRDU$
    C024    NERR3$
    C025    NERR4$
    C026    NIO3$
    C027    NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000026 1L       0000   000110 100F     0000   000222 10000F   0000   000260 10020F   0001   001575 10040L
    C001    001600 10045L   0002   000005 1216     0001   001072 1312L    0001   001120 1340L    0001   001243 1388L
    C001    001246 1390L    0000   000231 1399F    0001   001140 19L      0001   000276 20L      0000   000112 200F
    C001    000157 206G     0001   000177 2076     0001   001356 21L      0001   000257 2246     0001   000321 25L
    C001    000332 30L      0000   000124 300F     0001   000336 31L      0001   000472 3136     0001   000532 3246
    C001    000550 3306     0001   001462 35L      0001   000414 35L      0001   000655 40L      0000   000135 400F
    C001    001155 4376     0001   001173 4456     0001   001302 4736     0001   000064 5L       0001   000662 50L
    C000    000145 500F     0001   001313 5016     0001   001316 5046     0001   001371 5226     0001   001403 5276
    C001    001543 5646     0001   000666 60L      0000   000200 600F     0001   000414 70L      0001   000423 80L
    C000 I  000021 ADDRES   0004 I  000123 AREA     0000 I  000044 BEGIN1   0000 I  000045 BEGIN2   0003 I  000351 BESTVC
    C000 I  000024 BFINDX   0004 I  000115 BUFPIX   0000 I  000027 BUFSIZ   0004 I  000116 BUFTOT   0004 I  000062 CLSNAM
    C000 I  000072 CURADS   0003 I  000053 DATAPE   0003 I  000112 DIVSIZ   0004 I  000113 DUMP     0003 I  000062 DUPSYM
    C000 I  000041 ENDTAP   0003 I  000217 FETVC2   0003 I  000161 FETVEC   0003 I  000071 FILESV   0000 R  000052 FJ
    C000 I  000000 FL       0000 I  000267 FLDINF   0000 I  000034 FLDSAM   0003 I  000000 HEAD     0003 I  000255 HISVEC
    C000 I  000017 I        0000 I  000035 IB       0004 I  000000 IBEGIN   0003 I  000123 IBLOCK   0000 I  000022 IBUF
    C000 I  000031 ICK      0000 I  000036 IE       0000 I  000054 INADDR   0000    000311 INJP$    0000    000325 INJP$
    C000    000301 INJP$    0000    000321 INJP$    0000    000305 INJP$    0003 I  000313 INVERT   0004 I  000003 IPT
```

```
C004 I 000130 IQ1       0005 I 000226 ISUM      0005 I 000000 ITEST     0000 I 000137 IVALUE    0000 I 000023 IWRD
C000 I 000070 J         0000 I 000046 J         0000 I 000064 K         0000 I 000047 K         0003 I 000113 KEEPLV
C004 I 000107 KLBC      0000 I 000037 KNT       0000 I 000105 L         0010 I 000030 LAREAD    0000 I 000073 LASTWD
C004 I 000127 LBUF      0004 I 000121 LBUFD     0000 I 000040 LINE      0000 I 000270 LINEND    0000 I 000271 LININC
C000 I 000267 LINSTR    0000 I 000101 LL        0004 I 000102 LNCAT     0000 I 000014 LSTAT     0000 I 000076 LSTNPX
C003 I 000052 MAPTAP    0004 I 000122 MAXBF     0000 I 000057 MAXBUF    0003 I 000072 MAXCLS    0000 I 000026 MAXDIM
C003 I 000064 MAXDIV    0003 I 000055 MAXFET    0003 I 000074 MAXFLD    0003 I 000065 MINDIV    0005 I 000227 MSUM
C005 I 000144 MTEST     0000 I 000104 N         0003 I 000025 NBUFS     0004 I 000117 NBUFSD    0004 I 000120 NDUMP
C000 I 000067 NDXRD     0000 I 000043 NI        0004   000106 NIT       0003 I 000056 NM1       0000 I 000100 NN
C000 I 000020 NOCL      0003 I 000073 NOCLS2    0003 I 000121 NOCLS3    0003 I 000130 NOFEAT    0003 I 000101 NOFET2
C003 I 000102 NOFET4    0004 I 000064 NOFLD     0003 I 000075 NOFLD2    0003 I 000076 NOFLD3    0003 I 000111 NOGRP
C003 I 000110 NOHIST    0003 I 000107 NOSPEC    0003 I 000077 NOTRFD    0000 I 000075 NPIXEL    0004 I 000126 NPTS
C000 I 000103 NPXM1     0000 I 000077 NP1       0000 I 000033 NSAMP     0005 I 000230 NSUM      0000 I 000102 NTEMP
C005 I 000062 NTEST     0000 I 000032 NV        0004 I 000125 NWDRS     0004 I 000124 NWDS      0000 I 000066 NWORDS
C000 I 000030 NXTCLS    0003 I 000106 OSTAT     0000 I 000055 OUTADD    0003 I 000056 PAGSIZ    0003 I 000122 PCTSZ
C004 I 000103 PRNT      0004 I 000111 PROUT     0003 I 000114 PRTLEV    0004 I 000110 PRTME     0000 I 000065 RADRES
C000 I 000273 SAMEND    0000 I 000274 SAMINC    0000 I 000042 SAMPS     0000 I 000272 SAMSTR    0003 I 000054 SAVTAP
C004 I 000114 SCRAM1    0000 I 000062 SECTSZ    0003 I 000367 SERIAL    0000 I 000071 SIZRO     0000 I 000061 SLICE
C003 I 000120 SPCBAS    0003 I 000066 SPLMAX    0000 I 000074 STATUS    0004 I 000005 SYM       0003 I 000057 TAPCHK
C003 I 000070 TAPESV    0000 R 000053 TEMP      0003 I 000063 THRSYM    0004 I 000112 TOTPIX    0000 I 000063 TOTSTS
C004 I 000001 TOTWRD    0003 I 000060 TRNSYM    0000 I 000060 TSLICE    0003 I 000361 XSTSYM    0003 I 000103 VARSIZ
C003 I 000104 VARSZ2    0003 I 000105 VARSZ4    0000 R 000051 X         0003 I 000116 XHGH      0003 I 000117 XLOW
C003 I 000106 XSIZ      0003 I 000115 YSIZ      0000 R 000050 Z         0016 R 000000 ZOR

0010C    C*                                                                              000005
0010C    C*     THIS SUBROUTINE COORDINATES THE ROUTINES TO READ FIELDS OF DATA         000005
0010C    C*     FROM THE IMAGE TAPE AND STORE IT ON A DRUM FILE FOR THE CLASY           000005
0010C    C*     SUBROUTINES.  RANDOM ACCESS ROUTINES ARE USED FOR DRUM I/O. (RENIT      000005
0010C    C*     AND RWRITE).                                                            000005
0010C    C*                                                                              000005
00101             SUBROUTINE READTP(ARRAY,TOP,LAST,IDATA,TOPID)                          000005
00103             IMPLICIT INTEGER (A-Z)                                                 000005
00104             DIMENSION ARRAY(1),FLDINF(6),IDATA(TOPID),FL(12),LSTAT(3)              000005
00104    C                                                                               000005
00105             INCLUDE COMBK1                                                         000005
00107             INCLUDE CMBK10                                                         000005
00107    C                                                                               000005
0011E             EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(4),SAMSTR),                    000005
0011E         *               (FLDINF(2),LINEND), (FLDINF(5),SAMEND),                    000005
0011E         *               (FLDINF(3),LININC), (FLDINF(6),SAMINC)                     000005
0011E    C*                                                                              000005
0011E    C*     RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.     000005
0011E    C*     THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.                       000005
0011E    C*                                                                              000005
0011E    C*     FIELD INFORMATION STORED AS FOLLOWS                                     000005
0011E    C*                                                                              000005
0011E    C*                                                                              000005
0011E    C*         ARRAY(1)   =FIRST FIELD NAME FOR THIS CLASS                         000005
0011E    C*              (2)   =NO. OF VERTICES FOR THIS FIELD  (NV)                    000005
0011E    C*              (3)-(3+NV*2)  = ACTUAL VERTEX NUMBERS                          000005
0011E    C                                                                            NEW000005
0011E    C                                                                            NEW000005
0011E    C                                                                            NEW000005
0011E    C                                                                            NEW000005
```

```
      COMMON /ITEST/ ITEST( 50), NTEST( 50), MTEST( 50),ISUM,MSUM,NSUM
C
C  INITIALIZE CONSTANTS TO CHECK SCRAMBLING
      DO 2 I=1,50
      ITEST(I) = 0
      MTEST(I) = 0
    2 NTEST(I) = 0
      ISUM = 0
      MSUM = 0
      NSUM = 0
      IF(NOCL.GT.0)GO TO 1
      CALL RINIT(IBEGIN,NWDS)
      CALL TAPHDR(DATAPE)
    1 CONTINUE
      ADDRES=IBEGIN
      IBUF=1
      IWRD=0
      LAST=0
      TOTWRD=0
      BFINDX=2001
      NBUFS=3
      MAXDIM=TOP-2000
      BUFSIZ= MAXDIM/(NBUFS*NOFEAT)  * NOFEAT
      NOFLD=0
      IPT=1
      IF(NOCL.GT.0)WRITE(6,500)NXTCLS
C*
C*    READ A FIELD DESCRIPTION FROM CARDS.
C*
    5 ICK = LAREAD(ARRAY(IPT),ARRAY(IPT+2),FLDINF,ARRAY(IPT+1) )
      IF(ICK.NE.0)GO TO 20
      IF(ICK.EQ.0)GO TO 30
      NV=ARRAY(IPT+1)
      NOFLD=NOFLD+1
      NSAMP=(SAMEND-SAMSTR)/SAMINC+1
      FLDSAM=0
      IB=IPT+2
      IE=IB+NV*2-1
      WRITE(6,600)NOFLD,ARRAY(IPT),NV,SAMINC,LININC,
     *           (ARRAY(I),I=IB,IE)
C*
C*    POSITION TAPE FOR THIS FIELD
C*
      CALL FLDINT(FLDINF,FETVEC,NOFEAT)
      KNT=0
      DO 10 LINE=LINSTR,LINEND,LININC
      CALL LINERD(IDATA,ENDTAP)
      IF(ENDTAP.EQ.-1)GO TO 80
C*
C*    FIND SAMPLE INTERSECTS FOR THIS LINE.  NI=NO. OF INTERSECTS
C*
      CALL FDLINT(ARRAY(IPT+2),NV,FL,LINE,SAMPS,NI)
C*
C*    STORE DATA ON THIS LINE INTO OUTPUT BUFFER
C*
      CALL BUFILL(ARRAY(BFINDX),BUFSIZ,NBUFS,IDATA,NSAMP,$35)
      FLDSAM=FLDSAM+SAMPS
```

```
      10 CONTINUE                                                              000243
         IPT = IPT + NV*2 + 2                                                  000243
         ARRAY(IPT)=KNT                                                        000250
         DO 15 I=1,6                                                           000257
         IPT=IPT+1                                                             000257
      15 ARRAY(IPT)=FLDINF(I)                                                  000263
         IPT=IPT+1                                                             000266
         IF(IPT+30 .GT.2000)GO TO 70                                           000270
         GO TO 5                                                               000274
C*                                                                             000274
C*       CLASS NAME CARD ENCOUNTERED - REREAD PREVIOUS CARD TO GET NAME        000274
C*                                                                             000274
      20 NOCL=NOCL+1                                                           000276
         IF(NOCL.GT.1)GO TO 25                                                 000300
         READ(30,100)NXTCLS                                                    000303
         WRITE(6,500)NXTCLS                                                    000311
         GO TO 5                                                               000317
      25 CLSNAM=NXTCLS                                                         000321
         READ(30,100)NXTCLS                                                    000322
         GO TO 31                                                              000330
C*       EMPTY LAST BUFFER AND RETURN TO PROCESS DATA FOR THIS CLASS.          000330
      30 CLSNAM=NXTCLS                                                         000332
         LAST=1                                                                000333
C*                                                                             000333
      31 CALL EMPTY(ARRAY(BFINDX),BUFSIZ,NBUFS,$35)                            000336
C     BUFTOT = NO. OF AVAIL WORDS IN SCRATCH AREA 'ARRAY'                 CH   000336
         BUFTOT = ( TOP- IPT + 1 )/NOFEAT * NOFEAT                             000347
C     BUFSIZ = 1/2 OF TOTAL WORDS ON FAST STORAGE DEVICE BUFFER (ARRAY)   CH   000347
         BUFSIZ = BUFTOT/2                                                     000356
C                                                                              000356
C                                                                              000356
         NBUFSD = 0                                                            000360
C     NOWDS = TOTAL NO. OF WORDS AVAIL IN FAST STORAGE                         000360
C     TOTWRD = TOTAL NUMBER OF WORDS IN ORIG DATA ON DRUM                 CH   000360
C     SCRAM1 = 1ST WORD OF AVAIL FAST STORAGE + LENGTH OF ORIG DATA UNLESS CH  000360
         SCRAM1 = IBEGIN + TOTWRD                                              000361
C           SCRAMBLE THE INPUT DATA , PLACE THE SCRAMBLED DATA ON DRUM,        000361
C           FOR SUBSEQUENT ACCESS BY SUBROUTINES STATIS AND CLASY1             000361
C                                                                              000361
C     IPT = 1ST AVAIL WORD IN SCRATCH AREA 'ARRAY'                             000361
     340 BEGIN1 = IPT                                                          000364
C     BUFSIZ = 1/2 OF TOTAL AVAIL WORDS IN BUFFER 'ARRAY'                      000364
         BEGIN2 = BEGIN1 + BUFSIZ                                              000366
C                                                                              000366
C                                                                              000366
C     ** CALL SUBROUTINE SCRAM TO READ ORIG DATA, SCRAMBLE IT AND         CH   000366
C                                 REWRITE IT ON FAST STORAGE DEVICE **    CH   000366
         CALL SCRAM(  ARRAY(BEGIN1) , ARRAY(BEGIN2)  )                         000370
C                                                                              000370
C     BUFPIX = SIZE OF 'ARRAY'/NO OF CHANNELS                             CH   000370
         BUFPIX = BUFTOT/NOFEAT                                                000404
C                                                                              000404
C                                                                              000404
         RETURN                                                                000410
C                                                                              000410
C                                                                              000410
C                                                                              000410
```

```
35     CONTINUE
    70 WRITE(6,300)
       CALL CHERR
    80 WRITE(6,400)
       CALL CHERR
   100 FORMAT(10X,A6)
   200 FORMAT(' TOO MUCH DATA REQUESTED-DRUM SIZE IS',I10,' WORDS')
   300 FORMAT(' FIELD DEFINITION INFORMATION EXCEEDS 2000 WORDS')
   400 FORMAT(' END-OF-TAPE REACHED BEFORE END OF FIELD')
   500 FORMAT(//40X,'FIELDS TO BE CLUSTERED FOR CLASS',1X,A6//
     *  36X,'SAMPLE',3X,'LINE'/5X,'FIELD NAME',3X,
     *  'NO. OF VERTICES',3X,'INC.   ',3X,'INC.',33X,'VERTICES')
   600 FORMAT(1X,I2,4X,A6,12X,I2,10X,I2,6X,I2,5X,
     *  5('(',I4,',',I4,')',2X)/2(52X,5('(',I4,',',I4,')',2X)/))
C*
C*     INTERNAL ROUTINE BUFILL
C*
       SUBROUTINE BUFILL(CBUF,BUFSIZ,NBUFS,IDATA,NSAMP,$)
       DIMENSION CBUF(BUFSIZ,NBUFS), IDATA(NSAMP,NOFEAT)
       REAL CBUF
       DO 60 I=1,NI,2
       IB =(FL(I)-SAMSTR)/SAMINC + 1
       IE =(FL(I+1)-SAMSTR)/SAMINC + 1
       IF(MOD(SAMSTR,SAMINC) .NE. MOD(FL(I),SAMINC))IB=IB+1
       IF(IB .GT. IE)GO TO 60
       DO 50 J=IB,IE
       KNT=KNT+1
       DO 50 K=1,NOFEAT
       IWRD=IWRD + 1
       CBUF(IWRD,IBUF) = IDATA(J,K)
       IF(IWRD.LT.BUFSIZ)GO TO 50
       TOTWRD=TOTWRD + IWRD
       IF(TOTWRD.GT.NWDS)RETURN 6
       CALL RWRITE(ADDRES,CBUF(1,IBUF),BUFSIZ,LSTAT(IBUF))
C  COUNT OCCURANCES OF VALUES FOR TEST OF SCRAMBLING
       CALL TEST (CBUF(1,IBUF),BUFSIZ,ITEST,ISUM)
       IBUF=IBUF+1
       ADDRES=ADDRES + BUFSIZ
       IF(IBUF.GT.NBUFS)IBUF=1
    40 IF(LSTAT(IBUF).EQ.1)GO TO 40
       IWRD=0
    50 CONTINUE
    60 CONTINUE
       RETURN
C*
C*     INTERNAL ROUTINE EMPTY
C*
       SUBROUTINE EMPTY(CBUF,BUFSIZ,NBUFS,$)
       REAL CBUF
       DIMENSION CBUF(BUFSIZ,NBUFS)
       TOTWRD=TOTWRD+IWRD
       IF(TOTWRD.GT.NWDS)RETURN 4
       CALL RWRITE(ADDRES,CBUF(1,IBUF),IWRD,LSTAT(IBUF))
       CALL TEST (CBUF(1,IBUF),IWRD   ,ITEST,ISUM)
       RETURN
C
C**************************************************************************
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C
C         INTERNAL SUBROUTINE  SCRAM
C
C*****************************************************************************
C
C         PURPOSE:  SCRAMBLE THE ORDER OF A SET OF INTEGERS, IN THE
C                   RANGE  1 - NPIXEL , AND USE THIS SCRAMBLED SET OF
C                   INTEGERS TO SCRAMBLE THE LOCATIONS OF INPUT DATA
C                   WITHIN THE INPUT DATA BUFFER.  OUTPUT THE SCRAMBLED
C                   DATA ON THE DRUM.
C
C
C*****************************************************************************
C
C
      SUBROUTINE  SCRAM ( NUM , PIXEL )
      REAL   Z  ,  ZOR  ,  X  ,  FJ , PIXEL , TEMP
      DIMENSION  NUM(1)  ,  PIXEL(1)
C
C
      WRITE (6,10000) ISUM, (ITEST(I),I=1, 50)
10000 FORMAT (' ISUM=',I20,' ITEST=', /,(10I8))
C
C              ***   INITIALIZE  ***
C
      Z = ZOR( 145927 )
C
C
C          READ THE INPUT PIXELS FROM DRUM INTO THE BUFFER SPACE, AND
C          SCRAMBLE THE PIXELS IN THE INPUT BUFFER
C
C  INADDR = NEXT WORD OF ORIG DATA                                     CH
      INADDR = IBEGIN
C  OUTADD = NEXT AVAIL WORD FOR SCRAMBLED DATA                         CH
      OUTADD = SCRAM1
C  NM1 = NO OF CHANNELS - 1                                            CH
      NM1 = NOFEAT - 1
C
C
C
C             *** CALCULATE TRIAL SLICE  ***
C
C  NBUFS   = NO. OF BUFFERS OF DATA
C  TOTWRD  = TOTAL WORDS OF DATA
C  FEAT    = NO OF CHANNELS
C  MAXBUF  = MAXIMUM BUFFER SIZE * NO. OF BUFFERS
C  SLI'E   = LARGEST CHUNK THAT WILL FIT IN A BUFFER, NBUFSD TIMES
C  SEC.ION = ARRAY CREATED FROM SLICES OF DATA FROM EACH BUFFER
C
C  CALC TRIAL SLICE
1300  NBUFS = (TOTWRD + BUFSIZ -1) / BUFSIZ
1312  MAXBUF = BUFSIZ * NBUFS
      TSLICE = (MAXBUF/NBUFS) / NBUFS
C  SLICE MUST BE EVEN MULT OF NO OF CHANNELS
      IF (TSLICE .GE. NOFEAT) GO TO 1340
C
      WRITE (6,1399) TOTWRD, NBUFSD, NOFEAT, TSLICE
```

```
00426     259*     1399  FORMAT (' READTP--ERROR IN CALC BUFFER SLICES, TOTAL WORDS =',I1,                001115
00424     260*          1 ' NUMBER BUFFERS =',I8,' NUMBER OF CHANNELS =',I8,                            001115
00424     261*          2 ' TRIAL SLICE =',I8)                                                          001115
00425     262*           STOP                                                                           001115
00425     263*     C                                                                                    001115
00425     264*     C  CALC SLICE AS EVEN MULT OF NO CHANNELS                                            001115
00426     265*     1340  SLICE = (TSLICE/NOFEAT) * NOFEAT                                             001120
00426     266*     C  TOTAL WORDS READ = ( 1 SLICE FROM N BUFFERS)  N BUFFERS TIMES                     001120
00427     267*           SECTSZ = SLICE * NBUFS                                                         001124
00430     268*           TOTSTS = SECTSZ * NBUFS                                                        001126
00430     269*     C                                                                                    001126
00430     270*     C  NUMBER OF WORDS IN EACH BUFFER * NO OF BUFFERS MUST BE .GE. TOT WDS               001126
00431     271*           IF (TOTSTS .GE. TOTWRD) GO TO 19                                               001130
00433     272*           NBUFS = NBUFS + 1                                                              001133
00434     273*           GO TO 1312                                                                     001136
00434     274*     C                                                                                    001136
00434     275*     C                                                                                    001136
00435     276*      19   CONTINUE                                                                       001140
00435     277*     C                                                                                    001140
00435     278*     C  READ N BUFFERS OF DATA                                                            001140
00436     279*           DO  40  K=1,NBUFS,1                                                            001140
00436     280*     C                                                                                    001140
00436     281*     C                                                                                    001140
00436     282*     C                *** READ SLICES OF DATA  ***                                        001140
00436     283*     C                                                                                    001140
00436     284*     C  SLICE DATA--READ SOME DATA FROM EACH SECTION EXCEPT POSSIBLY LAST                  001140
00436     285*     C  SECTION.  START EACH READ IN FIRST SECTION                                        001140
00436     286*     C                                                                                    001140
00436     287*     C  INADDR = NEXT WORD OF ORIG DATA                                          CH       001140
00436     288*     C                                                                                    001140
00441     289*     1365  RADRES = INADDR + (K-1) * SLICE                                              001160
00442     290*           NWORDS = 0                                                                     001165
00442     291*     C                                                                                    001165
00443     292*           NDXRD = 1                                                                      001166
00444     293*           DO 1390 J = 1,NBUFS                                                            001173
00447     294*           SIZRD = SLICE                                                                  001176
00450     295*           CURADS = RADRES + (J-1) * SECTSZ                                               001200
00450     296*     C                                                                                    001200
00450     297*     C  CK IF MORE DATA IS NEEDED FROM LAST BUFFER                                        001200
00451     298*           IF (TOTWRD - CURADS  .LT. 0) GO TO 1390                                        001205
00451     299*     C                                                                                    001205
00452     300*           LASTWD = CURADS + SLICE - 1                                                    001210
00454     301*           IF (TOTWRD .LT. LASTWD) SIZRD = TOTWRD - CURADS + 1                            001213
00456     302*           CALL RREAD (CURADS, PIXEL(NDXRD), SIZRD, STATUS)                               001223
00457     303*           NDXRD = NDXRD + SIZRD                                                          001234
00460     304*           NWORDS = NWORDS + SIZRD                                                        001237
00461     305*     1388  IF (STATUS .GT. 0) GO TO 1388                                                001243
00461     306*     C  COUNT OCCURANCES OF VALUES FOR TEST OF SCRAMBLING                                 001243
00461     307*     1390  CONTINUE                                                                     001247
00463     308*     C                                                                                    001247
00463     309*     C                                                                                    001247
00463     310*     C         CONSTRUCT A SET OF SCRAMBLED INTEGERS IN THE RANGE 1 - NPIXEL              001247
00463     311*     C                                                                                    001247
00463     312*     C  CREATE SCRAMBLED INTEGERS ONLY WHEN BUFFER SIZE CHANGES                           001247
00465     313*     1395  CONTINUE                                                                     001247
00465     314*     C  NPIXEL = NO. OF SETS OF CHANNELS IN ONE BUFFER                           CH       001247
00466     315*           NPIXEL = NWORDS/NOFEAT                                                         001247
```

```
00467   316*          IF (K .GE. 2  .AND.  LSTNPX .EQ. NPIXEL) GO TO 21                    001253
00471   317*          LSTNPX = NPIXEL                                                       001267
00471   318*     C                                                                          001267
00472   319*          DO  5  I = 1,NPIXEL,1                                                 001271
00475   320*        5 NUM(I) = I                                                            001302
00475   321*     C                                                                          001302
00477   322*          NP1 = NPIXEL + 1                                                      001306
00477   323*     C                                                                          001306
00500   324*          DO  15  I=1,4,1                                                       001316
00500   325*     C                                                                          001316
00503   326*          DO  10  J=NPIXEL,1,-1                                                 001316
00506   327*          X = ZOR(0)                                                            001316
00507   328*          FJ = J                                                                001322
00511   329*          NN = FJ * X + 1.                                                      001325
00511   330*          LL = NP1 - J                                                          001336
00512   331*          NTEMP = NUM(LL)                                                       001341
00513   332*          NUM(LL) = NUM(NN)                                                     001345
00514   333*       10 NUM(NN) = NTEMP                                                       001351
00514   334*     C                                                                          001351
00516   335*       15 CONTINUE                                                            001356
00516   336*     C                                                                          001356
00516   337*     C                                                                          001356
00516   338*     C                    *** SCRAMBLE DATA ***                                 001356
00516   339*     C  NOPIXEL = NO. OF SETS OF CHANNELS IN ONE BUFFER                  CH     001356
00520   340*       21 NPXM1 = NPIXEL - 1                                                    001356
00521   341*          DO 30 I=1,NPXM1,2                                                     001360
00524   342*          N = NUM(I) * NOFEAT - NM1                                             001371
00526   343*          L = NUM(I+1) * NOFEAT - NM1                                           001374
00526   344*     C                                                                          001374
00526   345*     C  NOFEAT = NO. OF CHANNELS                                         CH     001374
00526   346*          DO  25  J=1,NOFEAT,1                                                  001403
00531   347*          NN = N + J - 1                                                        001403
00532   348*          LL = L + J - 1                                                        001411
00533   349*          TEMP = PIXEL(NN)                                                      001417
00534   350*          PIXEL(NN) = PIXEL(LL)                                                 001421
00535   351*       25 PIXEL(LL) =  TEMP                                                     001423
00537   352*       30 CONTINUE                                                              001431
00537   353*     C                                                                          001431
00537   354*     C                                                                          001431
00537   355*     C            *** WRITE SCRAMBLED DATA ON DRUM ***                          001431
00537   356*     C                                                                          001431
00537   357*     C        PUT THE BUFFER OF SCRAMBLED PIXELS BACK ON THE DRUM               001431
00537   358*     C                                                                          001431
00537   359*     C  OUTADD = NEXT AVAIL WORD FOR SCRAMBLED DATA                      CH     001431
00537   360*     C  PIXEL = SCRAMBLED DATA                                           CH     001431
00537   361*     C  NWORDS = NO. OF WORDS IN CURRENT BUFFER                          CH     001431
00541   362*          CALL RWRITE( OUTADD , PIXEL , NWORDS , OSTAT )                        001431
00541   363*     C  COUNT OCCURANCES OF VALUES FOR TEST OF SCRAMBLING                       001431
00542   364*          CALL TEST (PIXEL,NWORDS,NTEST,NSUM)                                   001437
00543   365*          WRITE (6,10020) NSUM, (NTEST(I),I=1, 50)                              001445
00547   366*     10020 FORMAT (' NSUM=',I20,' NTEST=', /,(10I8))                            001456
00550   367*          OUTADD = OUTADD + NWORDS                                              001456
00551   368*       35 IF ( OSTAT .GT. 0 )  GO TO 35                                         001462
00551   369*     C                                                                          001462
00553   370*       40 CONTINUE                                                              001466
00553   371*     C                                                                          001466
00555   372*          RETURN                                                                001466
```

```
00555      C                                                                     001466
00555      C                                                                     001466
00555      C        END OF INTERNAL SUBROUTUNE SCRAM                             001466
00555      C*****************************************************************    001466
00555      C                                                                     001466
00555      C                                                                     001466
00556            SUBROUTINE TEST (PIX,  NWORDS,LTEST,ISUM)                       001534
00561            REAL PIX                                                        001534
00562            DIMENSION PIX  (1), LTEST(1)                                    001534
00563            DO 10045 I=1,NWORDS                                             001534
00566            IVALUE = PIX  (I)                                               001543
00567            IF (IVALUE .LT. 1  .OR.  IVALUE .GT. 49) GO TO 10040            001551
00571            LTEST(IVALUE) = LTEST(IVALUE) + 1                               001566
00572            GO TO 10045                                                     001573
00573      10040 LTEST ( 50) = LTEST( 50) + 1                                    001575
00573      C                                                                     001575
00574      10045 ISUM = ISUM + IVALUE                                            001603
00576            RETURN                                                          001604
00577            END                                                             001636

END OF COMPILATION:      NO  DIAGNOSTICS.
```

```
@FOR,S SEPER,SEPER,SEPER
FOR SOE3-04/18/78-01:33:12 (0,0)

   SUBROUTINE SEPER     ENTRY POINT 000214

   STORAGE USED: CODE(1) 000237; DATA(0) 000045; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    CLPR
    C007    DENCAL
    C010    FREE
    C011    NPRT$
    C012    NIO1$
    C013    NIO2$
    C014    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000064 10L      0001    000041 20L      0001    000057 29L      0000    000006 571F     0001    000141 614L
    C000    000021 664F     0001    000067 666L     0003 R  000020 ALINK    0004    000064 AMM      0004    000075 AMOFAC
    C004    000077 AMOMAX   0004    000076 AMOMIN   0004    000100 AMORAT   0004    000054 AMQ      0004    000107 BETTER
    C004    000102 BIAS     0003 R  000027 CIN      0005    000001 CONLV    0004    000111 CORLEN   0003 R  000030 CTOT
    C003 P  000041 DCON     0004    000053 DELT     0003 R  000043 DISS     0004    000073 DWFAC    0004    000061 ELIMTH
    C004    000052 EPS      0003 R  000050 GEN      0004    000074 GRACTM   0003 R  000047 GREF     0003 I  000023 IOADJ
    C003 I  000024 INDEX    0004    000066 INDXVL   0000    000033 INJP$    0003 I  000000 JUNK     0000 I  000004 K
    C000 I  000005 KOLD     0004    000051 KROOT    0000 I  000000 KS       0003 I  000020 LINK     0004 I  000003 LKURT
    C004    000007 LOSUM    0004    000004 LOVAR    0004 I  000002 LR       0004 I  000006 LSKEW    0003 I  000021 LSUBS
    C004 I  000005 LSUM     0003 I  000022 LSUPER   0004 I  000005 LV       0004    000002 LVRIN    0005 R  000012 MACCEL
    C004    000001 MM       0004    000110 MODE     0004    000000 MQ       0004 I  000011 MXAR     0000 I  000001 N
    C003    000014 NARL     0004 I  000010 NINCLS   0004    000070 NPTSO    0003 I  000043 NSYMB    0003 I  000020 NTB
    C003    000016 NTBSZM   0003    000015 NTOP     0003    000017 NWANT    0003 R  000045 OCIN     0004    000055 ODCON
    C003 P  000047 ODEN     0003 R  000046 OPRIOR   0003 R  000035 OPROP     0003 R  000036 OW       0005 R  000010 PACCEL
    C000 R  000003 PASSF    0003 R  000050 PCOND    0003 R  000025 PCUM     0004    000103 PJOIN    0003 R  000043 PPASS
    C003 R  000042 PQRAT    0004    000071 PQRATH   0003 R  000026 PRIRCM    0003 R  000031 PROP     0003 R  000044 PST
    C000 P  000002 RATIO    0004    000065 SBLTH    0004    000062 SEPTH    0005    000002 SKBND    0005    000003 SKCHI
    C004    000112 SPCOR    0003 R  000032 SPFAC    0004    000072 SPMWTH   0005    000004 TRBND    0005    000005 TRCHI
    C005    000006 URKBND   0005    000007 URKCHI   0005 R  000014 VACCEL    0004    000063 VFAC     0003 R  000037 VOLIN
    C004    000101 VOLLIM   0003 R  000040 VOLRT    0003 R  000050 VRIN     0004    000104 VRJOIN   0003 R  000034 W
    C003 R  000033 WADJ     0004    000060 WADJIN   0005    000000 WAIT     0004    000106 WDELSM   0004    000067 WFAC
    C004    000105 WSIM     0004    000050 WTINIT   0004    000056 XOVFLO   0004    000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            SUBROUTINE SEPER(KL)                                                    000000
```

```
00101     2*      C  THIS ROUTINE IS CALLED WHENEVER IT HAS BEEN DECIDED THAT A                    000000
00101     3*      C     CLUSTER SHOULD BE SPLIT FOR GOOD.  THE CLUSTER HAS PREVIOUSLY              000000
00101     4*      C     BEEN SPLIT BY THE ROUTINE SPLIT, AND SUFFICIENT STATISTICS                 000000
00101     5*      C     HAVE NOW BEEN GATHERED TO CONFIRM THAT THE CLUSTER CAN BE                  000000
00101     6*      C     SPLIT UP ON A STATISTICALLY SIGNIFICANT BASIS.                             000000
00101     7*      C  THE ROUTINE TAKES THE CLUSTER AT KL, AND BRINGS UP ALL ITS                    000000
00101     8*      C     DAUGHTER CLUSTERS TO THE SAME LEVEL AS KL ITSELF.  KL IS                   000000
00101     9*      C     THEN ELIMINATED.                                                           000000
00101    10*            INCLUDE MISH                                                             000000
00101    11*      C-------------------------------------------------------------------------      000000
00101    12*      C                                                                               000000
00101    13*      C                        *** CALC NEW RATIO AND PASSF ***                       000000
00101    14*      C                                                                               000000
00131    15*            KS=LSUPER(KL)                                                            000000
00132    16*            N = LSUBS(KL)                                                            000002
00132    17*      C                                                                               000002
00133    18*            PRINT 571, INDEX(KL),INDEX(KS),INDEX(N),SPFAC(KL)                        000002
00141    19*       571  FORMAT ('0***SEPERATE ',I3,' SUPER,SUBS ',2I3,' SPFAC ',E11.5)           000020
00141    20*      C                                                                               000020
00142    21*            RATIO=PROP(KL)/PRIRCM(KL)                                                000020
00143    22*            PASSF=PST(KL)/(PCUM(KL)*PRIRCM(KL))                                      000023
00143    23*      C-------------------------------------------------------------------------      000023
00143    24*      C                                                                               000023
00143    25*      C                        *** REMOVE KL ***                                      000023
00143    26*      C                                                                               000023
00143    27*      C  K      = FIRST OFFSPRING OF PARENT OF KL                                     000023
00143    28*      C  FIRST WE FIND KL IN THE LIST OF LSUBS OF KS, AND REMOVE IT.                  000023
00148    29*            K=LSUBS(KS)                                                              000030
00149    30*            IF(K.NE.KL) GO TO 20                                                     000032
00149    31*      C  FIRST OFFSPRING OF KL = K, RESET FIRST OFFSPRING TO LINK(KL)                 000032
00147    32*            LSUBS(KS)=LINK(KL)                                                       000034
00150    33*            GO TO 29                                                                 000037
00150    34*      C                                                                               000037
00150    35*      C  FIND CLUSTER KL                                                              000037
00151    36*         20 KOLD=K                                                                    000041
00152    37*            K=LINK(K)                                                                000044
00153    38*            IF(K.LE.0) GO TO 666                                                     000046
00155    39*            IF(K.NE.KL) GO TO 20                                                     000050
00155    40*      C                                                                               000050
00155    41*      C  SET LINK OF KOLD = LINK OF KL                                                000050
00157    42*         25 LINK(KOLD)=LINK(KL)                                                       000053
00157    43*      C-------------------------------------------------------------------------      000053
00157    44*      C                                                                               000053
00157    45*      C                        CHECK FOR VOID SUBCLUSTERS OF KL                       000053
00157    46*      C                                                                               000053
00160    47*      C  PROCESS EACH SUBCLUSTER.                                                     000053
00161    48*         29 K=LSUBS(KL)                                                               000057
00162    49*            KOLD=KL                                                                  000061
00162    50*            10 CONTINUE                                                              000064
00163    51*            IF (K.GT.0) GO TO 614                                                    000064
00165    52*          666 PRINT 664,KL,K,KOLD                                                    000067
00172    53*          664 FORMAT('0BAD SUBINDEX IN SEPER: KL,K, K OLD=',2I6,I12)                 000100
00173    54*            CALL CLPR(KL,666,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                        000100
00174    55*            CALL CLPR(KOLD,666,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                      000115
00175    56*            RETURN                                                                   000135
00175    57*      C                                                                               000135
00175    58*      C                *** SET PARENT OF EACH OFFSPRING FROM PARENT OF KL             000135
```

```
00175   59*     C                                                                    000135
00176   60*       614 CONTINUE                                                       000141
00177   61*             LSUPER(K)=KS                                                 000143
00177   62*     C                                                                    000143
00177   63*     C  CALL DENCAL TO ADJUST THE DENOMINATOR OFFSET AND PROPORTION OF KL  000143
00200   64*             CALL DENCAL(K,RATIO,W(KL))                                   000145
00201   65*             PST(K)=PST(K)*PASSF                                          000154
00201   66*     C                                                                    000154
00201   67*     C  GET NEXT SIBLING                                                  000154
00202   68*             KOLD=K                                                       000160
00203   69*             K=LINK(K)                                                    000163
00204   70*             IF(K.NE.0) GO TO 10                                          000165
00204   71*     C                                                                    000165
00204   72*     C                   *** SET LAST OFFSPRING OF KL TO POINT TO OLD 1ST 000165
00204   73*     C                       OFFSPRING OF KL,S PARENT.                    000165
00204   74*     C                       SET KL,S PARENT TO POINT TO 1ST OFFSPRING OF KL ***  000165
00204   75*     C                                                                    000165
00204   76*     C  KS    = PARENT OF KL                                              000165
00204   77*     C  NOW ADD THE SUBCLUSTER LIST OF KL TO THAT OF KS                   000165
00206   78*             LINK(KOLD)=LSUBS(KS)                                         000167
00207   79*             LSUBS(KS)=LSUBS(KL)                                          000171
00210   80*             CALL FREE(KL,NINCLS)                                         000173
00211   81*             RETURN                                                       000177
00212   82*             END                                                          000236

     END OF COMPILATION:        1 DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S  SETUP9,SETUP9
FOR SOE3-03/23/78-10:41:42 (2,)

   SUBROUTINE SETUP9     ENTRY POINT 000271

   STORAGE USED: CODE(1) 000275; DATA(0) 000157; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    INFORM 000407
    0004    SUPCOM 000103
    0005    CLUSTR 000131

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    NXTCHR
    0007    NUMBER
    0010    NWDU$
    0011    NIO2$
    0012    NRDU$
    0013    NIO3$
    0014    NERR2$
    0015    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000014 10L      0001    000161 110L     0001    000172 130L     0001    000003 141G     0001    000202 150L
    0001    000041 162G     0001    000212 170L     0001    000220 180L     0001    000236 190L     0001    000260 260L
    0001    000100 30L      0000    000116 480F     0000    000120 490F     0001    000126 50L      0000    000130 500F
    0000    000132 510F     0000    000134 550F     0000    000137 630F     0001    000137 70L      0001    000150 90L
    0005    000123 AREA     0003    000351 BESTVC   0000 I  000107 BLANK    0005 I  000115 BUFPIX   0005 I  000116 BUFTOT
    0004 I  000005 CARD     0005 I  000002 CLSNAM   0000 I  000112 CODE     0004    000003 CODE1    0004    000004 CODE2
    0004 I  000002 COL      0003 I  000037 COMENT   0003    000053 DATAPE   0003 I  000016 DATE     0003    000112 DIVSIZ
    0005 L  000113 DUMP     0003    000062 DUPSYM   0003    000217 FETVC2   0003 I  000161 FETVEC   0003    000071 FILESV
    0003    000000 HEAD     0003 I  000002 HED1     0003 I  000023 HED2     0003    000255 HISVEC   0000 I  000111 I
    0005    000000 IBEGIN   0003    000123 IBLOCK   0000 I  000110 ICNT     0000    000151 INJP$    0004    000000 INTAPE
    0000 I  000075 INVEC    0003    000313 INVERT   0005    000003 IPT      0005    000130 IQ1      0000 I  000113 J
    0003    000113 KEEPLV   0005    000107 KLBC     0005    000127 LBUF     0005    000121 LBUFD    0005    000102 LNCAT
    0000 I  000115 M        0003    000052 MAPTAP   0005    000122 MAXBF    0003    000372 MAXCLS   0003    000064 MAXDIV
    0003    000055 MAXFET   0003    000074 MAXFLD   0003    000065 MINDIV   0005    000117 NBUFSD   0005    000120 NDUMP
    0005    000106 NIT      0003    000073 NOCLS2   0003    000121 NOCLS3   0003 I  000100 NOFEAT   0003    000101 NOFET2
    0003    000102 NOFET4   0005    000004 NOFLD    0003    000075 NOFLD2   0003    000076 NOFLD3   0003    000111 NOGRP
    0003    000110 NOHIST   0003    000107 NOSPEC   0003    000077 NOTRFD   0000 I  000114 NPOS     0005 I  000126 NPTS
    0007 I  000000 NUMBER   0005    000125 NWDRS    0005    000124 NWDS     0006 I  000000 NXTCHR   0003    000056 PAGSIZ
    0003    000122 PETSZ    0005 I  000103 PRNT     0005 I  000111 PROUT    0003    000114 PRTLEV   0005 I  000110 PRTME
    0003    000054 SAVTAP   0005 I  000114 SCRAM1   0003    000067 SERIAL   0000 I  000000 SMBLS    0003    000120 SPCBAS
    0003    000066 SPLMAX   0004    000001 STATUS   0005 I  000005 SYM      0003    000057 TAPCHK   0003    000070 TAPESV
    0003    000063 THRSYM   0005 I  000112 TOTPIX   0005 I  000001 TOTWRD   0003    000060 TRNSYM   0003    000061 TSTSYM
    0003 I  000103 VARSIZ   0003    000104 VARS22   0003    000105 VARS24   0003    000116 XHGH     0003    000117 XLOW
    0003    000106 XSIZ     0003    000115 YSIZ
```

```
00101            SUBROUTINE SETUP9
00101      C*******************************************************************      000000
00101      C*                                                                         000000
00101      C*    THE PURPOSE OF SUBROUTINE SETUP9 IS TO READ AND ANALYZE ALL CARD    000000
00101      C*    INPUT TO THE PROGRAM                                                 000000
00101      C*                                                                         000000
00101      C*******************************************************************      000000
00101      C*                                                                         000000
00103            IMPLICIT INTEGER (A-X)                                               000000
00103      C                                                                          000000
00104            INCLUDE COMBK1                                                    NEW000000
00106            INCLUDE COMBK4                                                    NEW000000
00111            INCLUDE COMBK8                                                    NEW000000
00113            INCLUDE CMBK10                                                    NEW000000
00113      C                                                                       NEW000000
00123            DIMENSION SMBLS(MAXPOP)                                           NEW000000
00124            DATA  SMBLS/'1','2','3','4','5','6','7','8','9','A','B','C','D',  -0000000
00124           *            'E','F','G','H','I','J','K','L','M','N','O','P','Q',     000000
00124           2            'R','S','T','U','V','W','X','Y','Z','$','#','A','/',     000000
00124                        '-','*','+','$','@','=','&','?','>','<',')','(',':',     000000
00124                        ''','\',';','*',1H ,',','.',' ','D' /                    000000
00126            DATA PRNT/ 2,1,3/                                                    000000
00126 *DIAGNOSTIC*  A LIST IN THE ABOVE STATEMENT IS TOO LONG.                       000000
00130            PARAMETER CNUM=10                                                    000000
00131            DIMENSION INVEC(CNUM)                                                000000
00132            DATA INVEC/'CHANNE','HED1','HED2','DATE','COMMEN','NPTS','NPOS' ,    000000
00132           * 'SYMBOL','PRINT','*END*'/                                           000000
00134            DATA BLANK/1H /,NPTS/10/                                             000000
00137            ICNT=0                                                               000000
00140            DO 5 I=1,MAXPOP                                                      000003
00143          5 SYM(I)=SMBLS(I)                                                      000003
00145            NOFEAT=0                                                             000005
00146            WRITE(6,630)                                                         000006
00150         10 READ(5,460)CODE,CARD                                                 000014
00154            WRITE(6,550) CODE,CARD                                               000024
00160            COL=0                                                                000035
00161            DO 20 I=1,CNUM                                                       000041
00164         20 IF(CODE .EQ. INVEC(I))GO TO(30,50,70,90,110,130,150,170,190,         000041
00164           * 260),I                                                              000041
00167            WRITE(6,490) CODE,CARD                                               000065
00173            GO TO 10                                                             000076
00173      C*                                                                         000076
00173      C     CHANNEL CARD                                                         000076
00173      C*                                                                         000076
00174         30 J = NXTCHR(CARD,COL)                                                 000100
00175            IF (J .EQ. BLANK)GO TO 10                                            000104
00177            COL=COL-1                                                            000106
00200            NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)                              000111
00201            VARSIZ=(NOFEAT*(NOFEAT+1))/2                                         000120
00202            GO TO 10                                                             000124
00202      C*                                                                         000124
00202      C*    HED1 CARD                                                            000124
00202      C*                                                                         000124
00203         50 READ (30,500)HED1                                                    000126
00206            GO TO 10                                                             000135
00206      C*                                                                         000135
```

```
00206     56*      C*     HED2 CARD                                                                  000135
00206     57*      C*                                                                                000135
00207     58*        70 READ (30,500) HED2                                                           000137
00212     59*           GO TO 10                                                                     000146
00212     60*      C*                                                                                000146
00212     61*      C*     DATE CARD                                                                  000146
00212     62*      C*                                                                                000146
00213     63*        90 READ(30,510)DATE                                                             000150
00216     64*           GO TO 10                                                                     000157
00216     65*      C*                                                                                000157
00216     66*      C*     COMMENT CARD                                                               000157
00216     67*      C*                                                                                000157
00217     68*       110 READ(30,500)COMENT                                                           000161
00222     69*           GO TO 10                                                                     000170
00222     70*      C*                                                                                000170
00222     71*      C*     NPTS CARD,NUMBER OF DATA POINTS FOR EACH CHANNEL RETURNED TO               000170
00222     72*      C*     CLASY3 EACH CALL TO CLASY2                                                 000170
00222     73*      C*                                                                                000170
00223     74*       130 J=NUMBER(CARD,COL,NPTS,0)                                                   000172
00224     75*           GO TO 10                                                                     000200
00224     76*      C*                                                                                000200
00224     77*      C*     NPOS CARD,NUMBER OF DRUM POSITIONS FROM WHICH TO OBTAIN DATA FOR           000200
00224     78*      C*     CLASY3,SO THAT THE DATA WILL BE SCRAMBLED                                  000200
00224     79*      C*                                                                                000200
00225     80*       150 J=NUMBER(CARD,COL,NPOS,0)                                                   000202
00226     81*           GO TO 10                                                                     000210
00226     82*      C*                                                                                000210
00226     83*      C*     SYMBOL CARD                                                                000210
00226     84*      C*                                                                                000210
00227     85*       170 ICNT=ICNT+1                                                                 000212
00230     86*           IF(ICNT .GT. MAXPOP) GO TO 10                                                000214
00232     87*       180 M=NXTCHR(CARD,COL)                                                          000220
00233     88*           IF( M .EQ. BLANK)GO TO 10                                                    000224
00235     89*           IF(M .EQ. ',')GO TO 180                                                      000227
00237     90*           SYM(ICNT)=M                                                                  000232
00240     91*           GO TO 170                                                                    000234
00240     92*      C*                                                                                000234
00240     93*      C      PRINT OPTION CARD                                                          000234
00240     94*      C*                                                                                000234
00241     95*       190 J=NXTCHR(CARD,COL)                                                          000236
00242     96*           IF (J .EQ. BLANK)GO TO 10                                                    000242
00244     97*           COL=COL-1                                                                    000244
00245     98*           J=NUMBER(CARD,COL,PRNT,0)                                                    000247
00246     99*           GO TO 10                                                                     000256
00246    100*      C*                                                                                000256
00246    101*      C*     *END* CARD                                                                 000256
00246    102*      C*                                                                                000256
00247    103*       260 RETURN                                                                      000260
00250    104*       480 FORMAT(A6,4X,62A1)                                                          000274
00251    105*       490 FORMAT(' INVALID INPUT CARD--IGNORED'/T5,A6,4X,62A1)                        000274
00252    106*       500 FORMAT(10X,10A6)                                                            000274
00253    107*       510 FORMAT(10X,2A6)                                                             000274
00254    108*       550 FORMAT(5X,A6,4X,62A1)                                                       000274
00255    109*       630 FORMAT(//' INPUT SUMMARY'//)                                                000274
00256    110*           END                                                                          000274
```

```
END OF COMPILATION:           1 DIAGNOSTICS.
```

```
@FOR,S SPLIT,SPLIT,SPLIT
FOR SOE3-04/18/78-01:33:20 (0,0)

   SUBROUTINE SPLIT     ENTRY POINT 003235

   STORAGE USED: CODE(1) 003416; DATA(0) 000435; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003   CLUS   002017
    C004   MISC   000113
    C005   STPAR  000016
    C006   CLUSTR 000131
    C007   SPPAR  000023

   EXTERNAL REFERENCES (BLOCK, NAME)

    C010   GET
    C011   SQMTX
    C012   EIGROT
    C013   MLT
    C014   MVEC
    C015   MTVEC
    C016   ACOM
    C017   APRIOR
    C020   MINV
    C021   SQRT
    C022   NWDU$
    C023   NIO2$
    C024   ACOS
    C025   COS
    C026   NPRT$
    C027   NIO1$
    C030   XPRI
    C031   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000304 103L      0001    000650 113L      0001    000367 118L      0001    000503 119L      0001    000726 150L
    C001    000120 166G      0001    001376 170L      0001    000161 176G      0001    002254 190L      0001    002336 200L
    C001    000173 203G      0001    000242 220G      0001    000273 235G      0001    000576 301G      0001    000603 305G
    C001    001001 341G      0001    001015 350G      0001    001151 367G      0001    001175 374G      0001    001571 460G
    C001    001747 510G      0001    002003 516G      0001    002212 537G      0001    002217 543G      0001    002277 560G
    C001    002315 567G      0001    002364 602G      0001    002365 605G      0001    002436 615G      0001    002445 620G
    C000    000264 653F      0001    003037 732G      0001    003071 743G      0000    000262 9999F     0000 R  000035 AA
    C003 R  000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX    0006    000076 AMOMIN
    C004    000100 AMORAT    0004 R  000054 AMQ       0000 R  000156 AMXVAL    0017 R  000000 APRIOR    0006    000123 AREA
    C000 R  000176 BB        0000 R  000202 BBP       0000 R  000172 BEST      0004    000107 BETTER    0004    000102 BIAS
    C000 R  000143 BQ        0000 R  000167 BTR       0006 I  000115 BUFPIX    0006 I  000116 BUFTOT    0003 R  000027 CIN
    C006 I  000002 CLSNAM    0005    000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT      0007 R  000013 DAMP
    C000 R  000163 DBES      0000 R  000250 DBYDSQ    0003 R  000041 DCON      0000 R  000235 DCOV2     0000 R  000044 DDEL
    C000 R  000247 DD3       0000 R  000224 DD5       0000 R  000004 DEL       0000 R  000164 DELFAC    0000 R  000150 DELIN
    C007 R  000001 DELMET    0000 R  000255 DELMOV    0000 R  000200 DELSQ     0004    000053 DELT      0000 R  000231 DERED
```

```
0000 P 000251 DGAM       0007 L 000015 DIAG       0003 R 000043 DISS       0000 R 000222 DKRTGM     0000 R 000221 DKURT
0000 R 000241 DKURT2     0000 R 000141 DOBFAC     0007 R 000014 DOBPMS     0000 R 000223 DSKEW      0000 R 000240 DSKEW2
0000 R 000054 DSQT       0006   000113 DUMP       0000 R 000227 DVDEL      0000 R 000234 DVD2D2     0004   000073 DWFAC
0000 R 000236 D2         0000 R 000237 D3         0000 R 000242 D5         0000 R 000243 D6         0000 R 000114 E
0000 R 000145 EE         0000 R 000165 EI         0004   000061 ELIMTH     0004   000052 EPS        0000 R 000205 ERCOV
0000 P 000276 ERED       0000 R 000207 ERKURT     0000 R 000206 ERSKEW     0000 R 000166 ERT        0000 R 000124 EVURT
0007 R 000010 EXMAX      0007 R 000006 EXMNSQ     0000 R 000215 EXPECT     0000 R 000162 FRT        0000 R 000147 GAM
0007 P 000011 GAMCEN     0000 R 000211 GAMCGN     0000 R 000204 GAMDEL     0007 R 000000 GAMMET     0000 R 000203 GAM2
0003 R 000050 GEN        0010 I 000000 GET        0000 R 000174 GM         0000 R 000212 GMCF       0000 R 000173 GP
0004   000074 GRACTM     0000 R 000137 GRADRT     0000 R 000136 GRADSQ     0003 R 000047 GREF       0000 I 000144 I
0006   000000 IBEGIN     0000 I 000151 IBES       0000 I 000134 IBLANK     0003 I 000023 IDADJ      0000 R 000002 IITEMP
0000 R 000003 IJTEMP     0003 I 000024 INDEX      0004 I 000166 INDXVL     0000   000344 INJPS      0006   000003 IPT
0006   000130 IQ1        0000 I 000135 ISTARS     0000 R 000001 ITEMP      0007 I 000022 ITER       0007 I 000020 ITERMX
0000 I 000146 J          0003 I 000000 JUNK       0000 I 000257 KA         0000 I 000260 KB         0006   000107 KLBC
0004   000051 KROOT      0006   000127 LBUF       0006   000121 LBUFD      0003 I 000020 LINK       0004   000003 LKURT
0006   000102 LNCAT      0000 I 000000 LOC        0004   000007 LOSUM      0004   000C04 LOVAR      0004 I 000002 LR
0004   000006 LSKEW      0003 I 000021 LSUBS      0004   000005 LSUM       0003 I 000022 LSUPER     0004 I 000005 LV
0004   000002 LVRIN      0005 R 000012 MACCEL     0006   000122 MAXBF      0004   000001 MM         0004   000119 NODE
0004 I 000000 MO         0000 I 000140 MQS        0004 I 000011 MXAR       0003   000C14 NARL       0006   000117 NBUFSD
0006   000120 NDUMP      0004 I 000010 NINCLS     0006   000106 NIT        0006   000004 NOFLD      0006   000126 NPTS
0004 I 000070 NPTSQ      0003 I 000043 NSYMB      0003 I 000020 NTB        0003   000016 NTBSZM     0003   000015 NTOP
0003   000017 NWANT      0006   000125 NWDRS      0006   000124 NWDS       0007 R 000003 OBCOV      0000 R 000213 OBEST
0000 R 000210 OBJ        0007 R 000005 OBKURT     0007 R 000004 OBSKEW     0003 R 000045 OCIN       0004 R 000055 ODCON
0003 R 000047 ODEN       0003 R 000046 OPRIOR     0003 R 000035 OPROP      0000 R 000153 ORTSM      0003 R 000036 OW
0005 R 000010 PACCEL     0003 R 000050 PCOND      0000 R 000214 PCTIMP     0003 R 000025 PCUM       0004   000103 PJOIN
0003 P 000043 PPASS      0003 R 000042 PQRAT      0004   000071 PQRATH     0003 R 000026 PRIRCM     0006 I 000103 PRNT
0003 R 000031 PROP       0006 I 000111 PROUT      0006 I 000110 PRTME      0003 R 000044 PST        0000 R 000276 R
0000 R 000157 RT         0000 R 000152 RTSM       0000 R 000104 S          0000 R 000161 SAB        0004   000065 SBLTH
0006 I 000114 SCRAM1     0004   000062 SEPTH      0007 R 000002 SGTMET     0000 R 000254 SGTMOV     0000 R 000244 SG1
0000 P 000216 SHRINK     0007 R 000007 SHRMIN     0000 R 000170 SIG        0005   000002 SKBND      0005   000003 SKCHI
0000 R 000217 SMOV       0004 R 000112 SPCOR      0003 R 000032 SPFAC      0004   000072 SPMVTH     0007 R 000021 SPRED
0000 R 000256 SPREDD     0000 R 000014 SPROA      0000 R 000171 SSIZ       0000 R 000252 SUMM       0000 R 000253 SUMV
0006 I 000005 SYM        0000 R 000024 T          0000 R 000245 TAU1       0000 R 000160 TCOF       0000 R 000226 TDEL
0000 R 000232 TEREDQ     0000 R 000220 THIMP      0007 R 000016 TIM0       0007 R 000017 TIM1       0000 R 000201 TMG
0006 I 000112 TOTPIX     0006 I 000001 TOTWRD     0000 R 000034 TPVD       0005   000004 TRBND      0005   000005 TRCHI
0000 R 000177 TRD        0000 R 000154 TRN        0000 R 000155 TRSQ       0000 R 000233 TR2VD4     0000 R 000230 TSPROA
0007   000012 TSQINI     0000 R 000225 TVDSQZ     0000 R 000246 UNIDSQ     0005   000006 URKBND     0005   000007 URKCHI
0005 R 000014 VACCEL     0000 P 000064 VDEL       0000 R 000074 VDSQD      0004   000063 VFAC       0003 R 000037 VOLIN
0004   000191 VOLLIM     0003 R 000040 VOLRT      0003 R 000050 VRIN       0004   000104 VRJOIN     0003 R 000034 W
0003 R 000033 WADJ       0004 R 000060 WADJIN     0005   000000 WAIT       0004   000106 WDELSM     0004 R 000067 WFAC
0000 P 000142 WO         0004   000105 WSIM       0000 R 000261 WSTART     0004   000050 WTINIT     0004   000056 XOVFLO
0004   000057 XUNFLO
```

```
00101          SUBROUTINE SPLIT(KL,SUM,SKEW,KURT,OSUM,OVAR,ORT,DSQ,                000051
00101         1   SG,TAU,ERE,VER,DUM,DSG,DTAU)                                    000051
00101          IMPLICIT REAL (A-H,O-Z)                                            000051
00101     C                                                                       000051
00101     C                                                                       000051
00101     C  THIS ROUTINE HAS THE FOLLOWING FUNCTIONS                             000051
00101     C     (1)  TO GUESS THE OPTIMAL AXIS TO SPLIT THE CLUSTER ON, USING     000051
00101     C          USING SKEWNESS AND KURTOSIS DATA.                            000051
00101     C     (2)  TO GERERATE TWO NEW CLUSTERS CORRESPONDING TO THE            000051
00101     C          PROBABLE HALVES OF THE OLD CLUSTER                           000051
00101     C     (3)  TO BUILD THEM INTO THE TREE                                  000051
```

```
      12*      C
      13*            REAL ITEMP, IITEMP, JJTEMP                                    000051
      14*            INCLUDE MISH                                                  000051
      15*            INCLUDE CMBK10                                                000051
      16*            REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)                   000051
      17*            REAL DSQ(MQ,MQ),SG(MQ,MQ),TAU(MQ,MQ),ERE(MQ,MQ),             000051
      18*           1  VER(MQ,MQ),ORT(MQ,MQ),DUM(MQ,MQ),DSG(MQ,MQ),DTAU(MQ,MQ)   000051
      19*      C                                                                  000051
      20*      C                                                                  000051
      21*      C  TO SAVE STORAGE, WE USE SEVERAL ARRAYS FROM THE CALLING         000051
      22*      C     SEQUENCE IN MORE THAN ONE WAY.  SINCE WE CANNOT EQUIVALENCE  000051
      23*      C     NAMES IN THE CALLING SEQUENCE, THESE ARRAYS HAVE NON-MNEMONIC000051
      24*      C     IDENTIFIERS.  IN PARTICULAR,                                 000051
      25*      C        DSQ IS ALSO USED AS THE TRANSPOSE OF ORT                  000051
      26*      C        ERE IS ALSO USED AS SG**2 AND AS D(OBJECTIVE)/D(SG**2)    000051
      27*      C        VER IS ALSO USED AS TAU**2 AND AS D(OBJECTIVE)/D(TAU**2)  000051
      28*      C        DUM IS A DUMMY ARRAY USED IN MANY WAYS                    000051
      29*      C                                                                  000051
      30*      C  THE ITERATION USES SG AND TAU AS THE SQUARE ROOT OF THE COVARIANCE000051
      31*      C  MATRIX, TO INSURE POSITIVE DEFINITENESS                         000051
      32*      C                                                                  000051
      33*      C  CONTROL PARAMETERS                                              000051
      34*      C                                                                  000051
      35*            COMMON/SPPAR/ GAMMET,DELMET,SGTMET,OBCOV,OBSKEW,OBKURT,EXMNSQ,000051
      36*           1     SHRPIN,EXMAX,GAMCEN,TSQINI,DAMP,DOBPMS,DIAG,TIM0,TIM1,ITERMX,000051
      37*           2        SPRED,ITER                                           000051
      38*            REAL DEL(MQMAX),SPROA(MQMAX),T(MQMAX),TPVD(MQMAX),DDEL     000051
      39*      C                                                                  000051
      40*      C  INTERNAL STORAGE                                                000051
      41*           1 (MQMAX),DSQT(MQMAX),VDEL(MQMAX),VDSQD(MQMAX),S(MQMAX),E(MQMAX),000051
      42*           2    EVURT(MQMAX),ERED(MQMAX)                                000051
      43*            REAL R(1)                                                   000051
      44*            EQUIVALENCE (ERED(1),R(1))                                  000051
      45*            LOGICAL DIAG                                                000051
      46*            DATA IBLANK,ISTARS /'      ','******'/                      000051
      47*            DATA GRADSQ,GRADRT/1.,1./                                   000051
      48*      C                                                                  000051
      49*            MQS=MQ*MQ                                                   000051
      50*      C  DOBFAC = MAXIMUM TIME,   TIM0,TIM1, DOBPMS IN COMMON CBLO       000051
      51*            DOBFAC=(TIM0+AMQ*TIM1)*DOBPMS                               000054
      52*      C                                                                  000054
      53*      C  WE MUST FIRST GENERATE CENTERED VERSIONS OF THE VARIANCE, SKEWNESS,000054
      54*      C     AND KURTOSIS.                                                000054
      55*      C                                                                  000054
      56*      C     VARIANCE                                                     000054
      57*      C                                                                  000054
      58*      C  EXPAND VRIN AND KURT                                            000054
      59*            WO=1E0/W(KL)                                                000061
      60*            CALL SQMTX(DUM,VRIN(KL+1))                                  000065
      61*            BQ=AMQ+2E0                                                  000073
      62*            CALL SQMTX(DTAU,KURT(KL+1))                                 000076
      63*      C                                                                  000076
      64*      C  REMOVE WEIGHT FACTOR FROM SKEW                                  000076
      65*            DO  61 I=1,MQ                                               000111
      66*      C     SKEWNESS=  SKEW                                              000111
      67*            S(I)=SKEW(I+KL)*WO                                          000120
      68*           61 CONTINUE                                                  000123
```

CH

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                                        000123
  120 CONTINUE                                                           000123
C    SHIFT TO FRAME WITH UNIT INVERSE COVARIANCE MATRIX (DUM).           000123
C  COORDINATE TRANSPATION CREATED IN ORT, EIGENVALUES IN E               000123
      CALL EIGROT(MQ,MQ,DUM,E,ORT)                                       000123
C                                                                        000123
C  DILATE ALONG COOPDINATE AXES TO MAKE COVAR A UNIT MATRIX              000123
      DO 101 I=1,MQ                                                      000132
      E(I)=E(I)*W(KL)                                                    000161
      EE=SQRT(ABS(E(I)))                                                 000163
C                                                                CH      000163
      DO 101 J=1,MQ                                                      000173
      ORT(J,I)=EE*ORT(J,I)                                               000173
  101 DSQ(I,J)=ORT(J,I)                                                  000175
C                                                                        000175
C  DO THE LINEAR TRANSFORMATIONS                                         000175
      CALL MLT(DUM,DTAU,ORT)                                             000205
      CALL MLT(DTAU,DSQ,DUM)                                             000212
      CALL MVEC(R,DSQ,S)                                                 000217
C                                                                        000217
C  THE PROBLEM IS NOW IN A FRAME WHERE THE COVAR MATRIX IS A UNIT MATRIX 000217
C                                                                        000217
  121 CONTINUE                                                           000224
C                                                                        000224
C  INITIALIZE AND MAKE GOOD INITIAL GUESS                                000224
      GAM=.000001E0                                                      000224
C                                                                CH      000224
C  FIND FRAME WHERE KURTOSIS IS DIAGONAL                                 000224
      CALL EIGROT(MQ,MQ,DTAU,EVURT,DUM)                                  000226
C   KURTOSIS=KURT-(MQ+2)*COVAR  (IN TRANSFORMED FRAME.)                  000226
      DO 68 I=1,MQ                                                       000235
   68 EVURT(I)=EVURT(I)*WO-BQ                                            000242
C                                                                        000242
C  ROTATE SKEWNESS TO THAT FRAME                                         000242
      CALL MTVEC(S,DUM,R)                                                000246
C                                                                        000246
C  INITIALIZATION                                                        000246
      DELIN=.3                                                           000253
      IBES=0                                                             000255
      RTSM=0.                                                            000256
      ORTSM=0.                                                           000257
      TRN=.05*AMQ                                                        000260
      TRSQ=TRN*TRN                                                       000263
      AMXVAL=0E0                                                         000265
C  FIND MAX NEGATIVE EIGENVALUE  CALC ROOT SUMS FOR SKEWNESS ADJUSTMENT  000265
      DO 111 I=1,MQ                                                      000273
      IF(EVURT(I).GT.AMXVAL) GO TO 103                                   000273
      AMXVAL=EVURT(I)                                                    000277
      IBES=I                                                             000301
  103 RT=SQRT(32.*AMAX1(0.,EVURT(I))+TRSQ)                               000304
      RTSM=RTSM+RT                                                       000317
  111 ORTSM=ORTSM+1./RT                                                  000321
      TCOF=4.*AMQ                                                        000327
C                                                                CH      000327
C  CK FOR NEG EIGENVALUE                                                 000327
      IF(IBES.EQ.0) GO TO 118                                            000332
C                                                                CH      000332
```

```
00025C     126*      C   NEGATIVE EIGENVALUE, ADJUST 'GOOD GUESS' TEMPORARIES                    000332
00025?     127*          DELIN=SQRT(SQRT(-8.*AMXVAL))                                         000334
00025?     128*          SAB=ABS(S(IBES))                                                     000347
00025?     129*          RTSM=RTSM+5.3333333*SAB/DELIN-TRN                                    000352
00025?     130*          ORTSM=ORTSM-1./TRN                                                   000357
00025?     131*          TCOF=TCOF+.333333                                                    000363
00025?     132*      C                                                                        000363
00025?     133*      C  POS AND NEG EIGENVALUE ADJUSTMENTS                                    000363
00025?     134*      C  CK FOR NEG EIGENVALUE                                                 000363
00025?     135*        118 TRN=TRN-(TCOF*TRN-RTSM)/(TCOF-TRN*ORTSM)                           000367
00026C     136*          TRSQ=TRN*TRN                                                         000377
00026?     137*      C                                                                        000377
00026?     138*          IF(IBES.EQ.0) GO TO 119                                              000401
00026?     139*      C                                                                        000401
00026?     140*      C   NEG EIGENVALUE, ADJUST 'GOOD GUESS' TEMPORARIES                     000401
00026?     141*          FRT=SQRT(-10.66667*AMXVAL)                                           000403
00026?     142*      C                                                                        000403
00026?     143*      C  THE COS,ACOS EXPPESSION FINDS THE ROOT OF A CUBIC                     000403
00026?     144*          ITEMP = (SAB*(4.*SAB-TRN*DELIN)/(AMXVAL*FRT))                        000412
00026?     145*          IITEMP = AMAX1(-.999999,ITEMP)                                       000423
00026?     146*          IJTEMP = AMIN1(.999999,IITEMP)                                       000432
00026?     147*          WRITE (6,9999) ITEMP, IITEMP, IJTEMP                                 000435
00027?     148*      9999  FORMAT (1X,6F16.6)                                                 000445
00027?     149*          DELIN = SAB/S(IBES) * AMIN1(2.,SQRT(FRT*COS(.33333333*               000445
00027?     150*         1    ACOS(IJTEMP))))                                                 000445
00027?     151*          DBES=2.*S(IBES)/DELIN-.5*TRN                                         000473
00027?     152*      C  IN ANY CASE, CREATE FACTOR USED IN MEAN DISPLACEMENT CALC             000473
00027?     153*        119 DELFAC=DELIN**6*OBKURT/OBSKEW**.5+2.5/SQRT(ABS(TRN)/AMO)           000503
00027?     154*      C                                                                        000503
00027?     155*      C                                                                        000503
00027?     156*      C  GENERATE ACTUAL INITIAL VALUES                                        000503
00030C     157*          DO 115 I=1,MQ                                                        000525
00030?     158*          EI=1./E(I)                                                           000576
00030?     159*      C                                                                        000576
00030?     160*      C  INITIAL COVARIANCE MATRICES AND ROTATION MATRICES                     000576
00030?     161*          DO 112 J=1,MQ                                                        000603
00030?     162*          DSG(J,I)=EI*OPT(J,I)                                                 000603
00031C     163*          SG(J,I)=DEO                                                          000605
00031?     164*        112 TAU(J,I)=DEO                                                       000606
00031?     165*      C                                                                        000606
00031?     166*      C   CALCULATE MEAN DISPLACEMENT USING SKEWNESS                          000606
00031?     167*          ERT=(SQRT(AMAX1(DEO,32.*EVURT(I)+TRSQ))-TRN)*.25                     000610
00031?     168*          BTR=2.*ERT+TRN                                                       000626
00031?     169*          DEL(I)=4.*S(I)*BTR/(DELFAC+BTR*BTR)                                  000631
00031?     170*          IF(I.NE.IBES) GO TO 113                                              000640
00031?     171*      C                                                                        000640
00031?     172*      C   SPECIAL CALCULATION ALONG MAX NEG EIGENVECTOR                        000640
00032C     173*          ERT=DBES                                                             000643
00032?     174*          DEL(I)=DELIN                                                         000645
00032?     175*        113 SIG=ABS(IEO-.25EO*DEL(I)*DEL(I))                                   000650
00032?     176*      C                                                                        000650
00032?     177*      C  CALCULATE COVARIANCE MATRIX DIAGONALS                                 000650
00032?     178*          SG(I,I)=SQRT(AMAX1(0.,AMIN1(2.*SIG-.001,SIG+ERT)))                   000655
00032?     179*        115 TAU(I,I)=SQRT(AMAX1(SIG-ERT,.001))                                 000676
00032?     180*          CALL MLT(ORT,DSG,DUM)                                                000713
00032?     181*      C                                                                        000713
00032?     182*      C  INITIALIZE ITERATIONS                                                 000713
```

```
00327           SSIZ=-.08E0                                                       000720
00330           BEST=1D30                                                         000722
00331           ITER=0                                                            000724
00331     C                                                                       000724
00331     C     ITERATION CYCLE STARTS HERE.                                      000724
00331     C                                                                       000724
00331     C       CALCULATE OBJECTIVE FUNCTION.                                   000724
00331     C                                                                       000724
00331     C    TEMPORARIES DEPENDING ON GAM                                       000724
00331     C                                                                       000724
00332       150 GP=.5E0*(1E0+GAM)                                                 000726
00333           GM=.5E0-.5E0*GAM                                                  000731
00334           AA=GM*GP                                                          000735
00335           BB=1.5E0*GAM*GAM-.5E0                                             000737
00335     C                                                                       000737
00335     C  CALC SIGMA SQ., TAU SQ AND DIFFERENCE (DSQ)                          000737
00336           CALL MLT(ERE,SG,SG)                                               000744
00337           CALL MLT(VER,TAU,TAU)                                             000751
00337     C                                                                       000751
00337     C  NOTE--MQS CAUSES PROCESSING OF WHOLE ARRAY                            000751
00340           DO 162 I=1,MQS                                                    000760
00343       162 DSQ(I,1)=ERE(I,1)-VER(I,1)                                        001001
00343     C                                                                       001001
00343     C  CALC DEL**2, TRACE DSQ                                               001001
00345           TRD=0E0                                                           001004
00346           DELSQ=0E0                                                         001005
00347           DO 161 I=1,MQ                                                     001015
00352           DELSQ=DELSQ+DEL(I)*DEL(I)                                         001015
00353       161 TRD=TRD+DSQ(I,I)                                                  001020
00353     C                                                                       001020
00353     C  CALC DSQ*DEL, DSQ*DSQ                                                001020
00355           CALL MVEC(R,DSQ,DEL)                                              001024
00356           CALL MLT(DUM,DSQ,DSQ)                                             001031
00356     C                                                                       001031
00356     C  TEMPS FOR OBJECTIVE FUNC CALC                                        001031
00357           TMG=TRD-GAM*DELSQ                                                 001036
00360           BBP=BB*DELSQ-GAM*TRD                                              001043
00361           GAM2=2E0*GAM                                                      001051
00362           GAMDEL=GAM*DELSQ                                                  001054
00363           ERCOV=AMQ                                                         001056
00364           ERSKEW=0E0                                                        001060
00365           ERKURT=0E0                                                        001061
00365     C                                                                       001061
00365     C  VECTORS AND ARRAYS USED HERE ARE ALSO USED IN THE DERIVATIVE CALC    001061
00365     C  CALC ACTUAL ERRORS                                                   001061
00366           DO 165 I=1,MQ                                                     001151
00366     C     DELTA 3                                                           001151
00371           SPROA(I)=TRD*DEL(I)+2E0*R(I)-GAMDEL*DEL(I)                        001156
00372           T(I)=AA*SPROA(I)-S(I)                                             001167
00373           DO 166 J=1,MQ                                                     001175
00373     C     DELTA 2                                                           001175
00376           ERE(I,J)=AA*DEL(I)*DEL(J)+GP*ERE(I,J)+GM*VER(I,J)                 001175
00377           ERCOV=ERCOV+ERE(I,J)**2                                           001205
00377     C     DELTA 4                                                           001205
00400           VER(I,J)=AA*(TMG*DSQ(I,J)+2E0*DUM(I,J)+BBP*DEL(I)*DEL(J)-         001210
00400          1   GAM2*(DEL(I)*R(J)+DEL(J)*R(I)))                                001210
00400     C                                                                       001210
```

```
00400    240*      C  CALC ERRORS IN KURTOSIS(ERKURT), COVARIANCE(ERCOV), SKEWNESS(ERSKEW)          001210
00401    241*        166 ERKURT=ERKURT+VER(I,J)    **2                                           001231
00403    242*            ERCOV=ERCOV-2E0*ERE(I,I)                                                001235
00404    243*            ERKURT=ERKURT+(-2E0*VER(I,I)+EVURT(I))*EVURT(I)                         001241
00405    244*            VER(I,I)=VER(I,I)-EVURT(I)                                              001247
00406    245*            ERE(I,I)=ERE(I,I)-1E0                                                   001252
00407    246*        165 ERSKEW=ERSKEW+T(I)*T(I)                                                 001255
00407    247*      C     TEST NEW POINT                                                          001255
00407    248*      C                                                                             001255
00407    249*      C  CALC OBJECTIVE FUNCTION                                                    001255
00407    250*      C  OBCOV, OBSKW, OBKURT ARE USED AS PARAMETERS DEFINED IN CBLO                001255
00411    251*            OBJ=OBCOV*ERCOV+OBSKEW*ERSKEW+OBKURT*ERKURT                             001277
00412    252*            GAMCGN=GAM*GAMCEN                                                       001305
00413    253*            GMCF=1E0+GAM*GAMCGN                                                     001310
00414    254*            OBJ=OBJ*GMCF                                                            001313
00414    255*      C                                                                             001313
00414    256*      C  CALC STEP SIZE (SSIZ) AND                                                  001313
00415    257*            OBEST=BEST                                                              001315
00416    258*            IF(ITER.EQ.0) PCTIMP=.25                                                001317
00420    259*            EXPECT=SSIZ*GRADRT*GMCF                                                 001323
00421    260*            SHRINK=1.+(BEST-OBJ)/EXPECT                                             001327
00422    261*            SHRINK=AMAX1(.5E0/AMAX1(SHRINK,1E-10),SHRMIN)                           001334
00422    262*      C                                                                             001334
00422    263*      C  CK TO SEE IF OBJECTIVE FUNCTION HAS IMPROVED                               001334
00423    264*            IF(OBJ.LE.BEST) GO TO 170                                               001347
00423    265*      C  OBJ FUNCTION HAS NOT IMPROVED, SHRINK STEP SIZE, SKIP NEW DERIV CALC       001347
00425    266*            SMOV=(SHRINK-1E0)*SSIZ                                                  001353
00426    267*            SSIZ=SSIZ*SHRINK                                                        001356
00427    268*            THIMP=AMIN1(THIMP,OBJ)                                                  001361
00430    269*            PCTIMP=PCTIMP-PCTIMP*DAMP*.7                                            001367
00431    270*            GO TO 190                                                               001374
00431    271*      C                                                                             001374
00431    272*      C   OBJ FUNCTION IMPROVED, CONCLUDE STEP SIZE CALC                            001374
00432    273*       170  THIMP=AMIN1(BEST-OBJ,OBJ)                                             001376
00433    274*            PCTIMP=PCTIMP+(THIMP/OBJ-PCTIMP)*DAMP                                   001404
00434    275*            IF(PCTIMP*OBJ.LE.DOBFAC.OR.ITER.GT.ITERMX)   GO TO 200                  001411
00436    276*            BEST=OBJ                                                                001427
00437    277*            SHRINK=AMIN1(SQRT(EXMNSC+(1E0-SHRINK)**2),EXMAX)                        001431
00440    278*            SSIZ=SSIZ*SHRINK                                                        001446
00441    279*            SMOV=SSIZ                                                               001450
00441    280*      C                                                                             001450
00441    281*      C     CALCULATE DERIVATIVES                                                   001450
00441    282*      C   TEMP SCALARS DEPENDING ON OBKURT, OBSKEW DEFINITION                       001450
00442    283*            DKURT=AA*OBKURT                                                         001451
00442    284*            DKRTGM=DKURT*GAM                                                        001454
00444    285*            DSKEW=AA*OBSKEW                                                         001456
00445    286*            DD5=-2E0*DKRTGM                                                         001461
00445    287*      C   TEMP VECTORS AND MATRIX PRODUCTS                                        001461
00446    288*            CALL MVEC(ERED,ERE,DEL)                                                 001463
00447    289*            CALL MVEC(DSQT,DSQ,T)                                                   001470
00450    290*            CALL MVEC(VDEL,VER,DEL)                                                 001475
00451    291*            CALL ACOM(DUM,VER,DSQ)                                                  001502
00452    292*            CALL MVEC(VDSQD,DUM,DEL)                                                001507
00452    293*      C  INITIALIZE FOR INNER PRODUCTS                                              001507
00453    294*            IVDSQ2=0E0                                                              001514
00454    295*            TDEL=0E0                                                                001515
00455    296*            DVDEL=0E0                                                               001516
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      TSPROA=DEO
C     CALC. INNER PRODUCTS
      DO 171 I=1,MQ
      TDEL=TDEL+DEL(I)*T(I)
      DVDEL=DVDEL+DEL(I)*VDEL(I)
      TSPROA=TSPROA+T(I)*SPROA(I)
      TVDSQ2=TVDSQ2+DUM(I,I)
  171 TPVD(I)=DSKEW*T(I)+DD5*VDEL(I)
C     INITIALIZE FOR MORE INNER PRODUCTS
      DERED=DEO
      TEREDQ=DEO
      TR2VD4=DEO
      DVD2D2=DEO
      DCOV2=2EO*OBCOV*AA
C     CALC. DERIVATIVE COEFFICIENT TEMPORARIES
      D2=2EO*AA*(OBKURT*(BB*DVDEL-.5EO*GAM*TVDSQ2)-OBSKEW*GAM*TDEL)
      D3=DSKEW*(TRD-GAM*DELSQ)
      DSKEW2=2EO*DSKEW
      DKURT2=2EO*DKURT
      D5=DKURT2*BBP
      D6=-2EO*DKURT2*GAM
      SG1=OBCOV*GP
      TAU1=OBCOV*GM
      UNIDSQ=DSKEW*TDEL+DKURT*.5EO*TVDSQ2-DKRTGM*DVDEL
      DD3=DKURT*TRD-DKRTGM*DELSQ
C
C     CALC  MATRIX TEMPS AND DOT PRODUCTS
      DO 175 I=1,MQ
      DERED=DERED+DEL(I)*ERED(I)
      DVD2D2=DVD2D2+DEL(I)*VDSQD(I)
C
C     DDEL IS THE DERIVATIVE WITH RESPECT TO DEL
      DDEL(I)=DCOV2*ERED(I)+D2*DEL(I)+D3*T(I)+DSKEW2*DSQT(I)+D5*VDEL(I)+
     1  D6*VDSQD(I)
      DO 174 J=1,MQ
      TEREDQ=TEREDQ+ERE(I,J)*DSQ(I,J)
      TR2VD4=TR2VD4+DSQ(I,J)*DUM(I,J)
      DBYDSQ=TPVD(I)*DEL(J)+TPVD(J)*DEL(I)
     1        +DD3*VER(I,J)+DKURT2*DUM(I,J)
      VER(I,J)=TAU1*ERE(I,J)-DBYDSQ
  174 ERE(I,J)=SG1*ERE(I,J)+DBYDSQ
      ERE(I,I)=ERE(I,I)+UNIDSQ
  175 VER(I,I)=VER(I,I)-UNIDSQ
C     CALC DERIVATIVES WITH RESPECT TO COVARIANCE MATRIX ROOTS
      CALL ACOM(DSG,SG,ERE)
      CALL ACOM(DTAU,TAU,VER)
C     CALC DERIVATIVE WITH RESPECT TO GAM
      DGAM=OBCOV*(-.5EO)*(GAM*DERED-TEREDQ)-OBSKEW*(.5EO*GAM*TSPROA+
     1  AA*DELSQ*TDEL)-OBKURT*(GAM*(.25EO*TVDSQ2*TRD+.5EO*TR2VD4+
     2  (BB-.5EO)*DELSQ*DVDEL)-
     3  .5EO*BB*(.5EO*DELSQ*TVDSQ2+DVDEL*TRD+2EO*DVD2D2))+
     4  GAMCGN/(GMCF*GMCF)*OBJ
C     CALC THE SQ. OF THE DERIVATIVE AND ITS ROOT
      SUMM=DEO
      SUMV=DEO
      DO 181 I=1,MQ
      SUMV=SUMV+DDEL(I)*DDEL(I)
```

```
      DO 181 J=1,MQ
      SUMM=SUMM+DSG(I,J)*DSG(I,J)+DTAU(I,J)*DTAU(I,J)
  181 CONTINUE
      GRADSQ=SUMM*SGTMET+SUMV*DELMET+DGAM*DGAM*GAMMET
      GRADRT=SQRT(GRADSQ)
C
C  SET UP AND TEST POINT.
C  ENTRY FROM NO DERIVATIVE CALC.
  190 CONTINUE
  194 CONTINUE
C
C
C  MOVE TO NEW POINT
  195 SMOV=SMOV/GRADRT
      SGTMOV=SMOV*SGTMET
      DO 191 I=1,MQS
      SG(I,1)=SG(I,1)+SGTMOV*DSG(I,1)
  191 TAU(I,1)=TAU(I,1)+SGTMOV*DTAU(I,1)
      DELMOV=DELMET*SMOV
      DO 192  I=1,MQ
  192 DEL(I)=DEL(I)+DELMOV*DDEL(I)
      GAM=GAM+SMOV*GAMMET*DGAM
C
C  ITERATE AND LIMIT NUMBER OF ITERATIONS
      ITER=ITER+1
      IF (ITER.GT.ITERMX) GO TO 200
      GO TO 150
C
C
C  *** GENERATE TWO NEW SUBCLUSTERS **
C
C  SHIFT COORDINATE FRAME BACK
  200 CONTINUE
C
C  CALC DSQ = TRANSPOSE OF OLD OLD ROTATION
  250 DO 251 I=1,MQ
      DO 251 J=1,MQ
  251 DSQ(J,I)=ORT(I,J)
C  DSG AND DTAU ARE TEMP ARRAYS FOR COVARIANCES
      CALL MLT (DSG,SG,SG)
      CALL MLT(DTAU,TAU,TAU)
C
C  SHEAR THE MATRICES OUT BY THE ARBITRARY FACTOR 'SPRED' FROM CBLO
      DO 253 I=1,MQ
      DO 253 J=1,MQ
      SPREDD=.2*SPRED*DEL(I)*DEL(J)
      IF(I.EQ.J) SPREDD=SPREDD+SPRED
      DSG(I,J)=DSG(I,J)+SPREDD
  253 DTAU(I,J)=DTAU(I,J)+SPREDD
C
C  DO ACTUAL ROTATION
      CALL MLT(DUM,DSG,DSQ)
      CALL MLT(DSG,ORT,DUM)
      CALL MLT(DUM,DTAU,DSQ)
      CALL MLT (DTAU,ORT,DUM)
      CALL MVEC (R,ORT,DEL)
C
```

```
C  CREATE AND LINK NEW CLUSTERS
      KA=GET(NINCLS)
      KB=GET(NINCLS)
C
C  CREATE NAMES AND LINKAGES FOR NEW CLUSTERS KA, KB
      INDXVL=INDXVL+2
      INDEX(KA)=INDXVL-1
      INDEX(KB)=INDXVL
      LINK(KB)=0
      LSUBS(KB)=0
      LSUBS(KA)=0
      LINK(KA)=KB
      LSUBS(KL)=KA
      LSUPER(KA)=KL
      LSUPER(KB)=KL
C
C  IDADJ = ADJUSTMENT POSITION IN TERMS OF INPUT POINTS
      IDADJ(KA)=NPTSO+TOTPIX
      IDADJ(KB)=IDADJ(KA)
C
C  SET UP WEIGHTS AND PROPORTIONS
      PROP(KA)=GP
      PROP(KB)=GM
      OPROP(KA)=GP
      OPROP(KB)=GM
      SPFAC(KA)=-9999.
      SPFAC(KB)=-9999.
      PORAT(KA)=0.
      PORAT(KB)=0.
      PRIRCM(KL)=1.
      SPFAC(KL)=APRIOR(KL)
      OPRIOR(KL)=SPFAC(KL)
C  SET PARAMETERS.
      WSTART=WFAC*AMQ*SPCOR
      W(KA)=WSTART
      OW(KA)=W(KA)
      CIN(KA)=WSTART*PROP(KA)
      OCIN(KA)=CIN(KA)
      ODEN(KA)=CIN(KA)/GP
      CTOT(KA)=W(KL)-ODEN(KA)
      W(KB)=WSTART
      OW(KB)=W(KB)
      CIN(KB)=WSTART*PROP(KB)
      OCIN(KB)=CIN(KB)
      ODEN(KB)=CIN(KB)/GM
      CTOT(KB)=W(KL)-ODEN(KB)
      WADJ(KA)=W(KA)+WADJIN
      WADJ(KB)=W(KB)+WADJIN
C
C  INVERT COVAR MATRIX AND CALC VOLUME
      CALL MINV(SG,DUM,DSG,VOLIN(KA))
      CALL MINV(TAU,DUM,DTAU,VOLIN(KB))
      IF(VOLIN(KA).LE.0..OR.VOLIN(KB).LE.0.) PRINT 653,KL,KA,KB,
     1     VOLIN(KL),VOLIN(KA),VOLIN(KB)
  653 FORMAT('0VOLUME ERROR IN SPLIT: CLASSES, VOLUMES',3I5,3E10.5)
      VOLIN(KA)=ABS(VOLIN(KA))*.8756510763E-26*(6.283185307)**MQ
      VOLIN(KB)=ABS(VOLIN(KB))*.8756510763E-26*(6.283185307)**MQ
```

```
00724     468*            VOLRT(KA)=SQRT(VOLIN(KA))                                    002756
00725     469*            VOLRT(KB)=SQRT(VOLIN(KB))                                    002764
00726     470*            DCON(KA)=ODCON                                               002772
00727     471*            DCON(KB)=ODCON                                               002774
00730     472*            LOC=0                                                        002776
00730     473*      C                                                                  002776
00730     474*      C  SET UP ALL THE ARRAYS AND VECTORS FOR NEW CLUSTER               002776
00731     475*            DO 210 I=1,MQ                                                003037
00734     476*            SKEW(KA+I)=0.                                                003042
00735     477*            SKEW(KB+I)=0.                                                003043
00736     478*            SUM(KA+I)=WSTART*(SUM(I+KL)/W(KL)+GM*R(I))                   003044
00737     479*            SUM(KB+I)=WSTART*(SUM(I+KL)/W(KL)-GP*R(I))                   003053
00740     480*            OSUM(KA+I)=SUM(KA+I)                                         003062
00741     481*            OSUM(KB+I)=SUM(KB+I)                                         003064
00742     482*            DO 210 J=1,I                                                 003071
00742     483*      C                                                                  003071
00742     484*      C  LOC IS A LOCAL INDEX WITHIN TRIANGULAR ARRAYS                   003071
00745     485*            LOC=LOC+1                                                    003076
00746     486*            VRIN(KA+LOC)=SG(I,J)/WSTART                                  003123
00747     487*            VRIN(KB+LOC)=TAU(I,J)/WSTART                                 003135
00750     488*            CURT(KA+LOC)=0.                                              003152
00751     489*            CURT(KB+LOC)=0.                                              003155
00752     490*            OVAR(KA+LOC)=DSG(I,J)*WSTART                                 003157
00753     491*          210 OVAR(KB+LOC)=DTAU(I,J)*WSTART                              003164
00756     492*            RETURN                                                       003204
00757     493*            END                                                          003415

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,S SQMTX,SQMTX,SQMTX
FOR SQE3=04/18/78-01:34:29 (0,0)

   SUBROUTINE SQMTX     ENTRY POINT C00074

   STORAGE USED: CODE(1) 000105; DATA(0) 000030; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS   002017
    C004    MISC   000113
    C005    STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000015 1356      0001    000032 1416      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    C004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER
    C004    000102 BIAS      0003 R  000027 CIN       0005    000001 CONLW     0004    000111 CORLEN    0003 R  000030 CTOT
    C003 R  000041 DCON      0004    000053 DELT      0003 R  000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH
    C004    000052 EPS       0003 R  000050 GEN       0004    000074 GRACTM    0003 R  000047 GREF      0000 I  000002 I
    C003 I  000023 IDADJ     0000 I  000003 IJ        0000 I  000001 IMQ       0003 I  000024 INDEX     0004    000066 INDXVL
    C000    000006 INJP$     0000 I  000004 J         0003 I  000000 JUNK      0004    000051 KROOT     0003 I  000020 LINK
    C004    000003 LKURT     0000 I  000000 LOC       0004    000007 LOSUM     0004    000004 LOVAR     0004 I  000002 LR
    C004    000006 LSKEW     0003 I  000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV
    C004    000002 LVRIN     0005 R  000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I  000000 MQ
    C004 I  000011 MXAR      0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO     0003 I  000043 NSYMB
    C003 I  000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT     0003 R  000045 OCIN
    C004    000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW
    C005 R  000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS
    C003 R  000042 PORAT     0004    000071 PORATH    0003 R  000026 PRIRCM    0003 R  000031 PROP      0003 R  000044 PST
    C004    000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR
    C003 P  000032 SPFAC     0004    000072 SPMVTH    0005    000004 TRBND     0005    000005 TRCHI     0005    000006 URKBND
    C005    000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN     0004    000101 VOLLIM
    C003 R  000040 VOLRT     0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R  000034 W         0003 R  000033 WADJ
    C004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIM
    C004    000050 WTINIT    0004    000056 XOVFLO    0004    000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*          SUBROUTINE SQMTX(SQ,AM)                                                    000010
00101      2*      C  THIS SUBROUTINE EXPANDS MATRIX AM FROM TRIANGULAR FORM AND MAKES            000010
00101      3*      C    AN MQ*MQ SQUARE SYMMETRIC MATRIX IN SQ(DIM MQ*MQ).                        000010
00102      4*            INCLUDE MISH                                                             000010
00131      5*            DIMENSION AM(475),SQ(900)                                                000010
00132      6*            LOC=0                                                                    000010
00133      7*            IMQ=0                                                                    000011
00134      8*            DO 11 I=1,MQ                                                             000015
```

```
00137      9*            IJ=I                                    000024
00140     10*            DO 10 J=1,I                             000032
00143     11*            LOC=LOC+1                               000033
00144     12*            SQ(IJ)=AM(LOC)                          000036
00145     13*            SQ(IMQ+J)=AM(LOC)                       000042
00146     14*         10 IJ=IJ+MQ                                000044
00150     15*         11 IMQ=IMQ+MQ                              000050
00152     16*            RETURN                                  000055
00153     17*            END                                     000104
```

END OF COMPILATION: 1 DIAGNOSTICS.

```
@FOR,S STATIS,STATIS,STATIS
FOR SOE3-04/18/78-01:34:33 (0,0)

   SUBROUTINE STATIS    ENTRY POINT 001252

   STORAGE USED: CODE(1) 001305; DATA(0) 000306; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS   002017
    C004    MISC   000113
    C005    STPAR  000016
    C006    CLUSTR 000131
    C007    RAND   000003

   EXTERNAL REFERENCES (BLOCK, NAME)

    C010    DISC
    C011    RREAD
    C012    CORECT
    C013    DOTSQ
    C014    VPV
    C015    VMTV
    C016    MPVS
    C017    ADJUST
    C020    CLDUMP
    C021    NWDU$
    C022    NIO2$
    C023    EXP
    C024    NPRT$
    C025    NIO1$
    C026    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001      000011 1L        0001      000075 10L       0000      000137 1000F     0001      000157 130L      0001      000171 131L
    C001      000465 132L      0001      000230 133L      0001      000255 134L      0001      000307 138L      0001      000505 140L
    C001      000534 143L      0001      000557 144L      0001      000435 151L      0001      000451 153L      0001      001044 163L
    C001      001055 164L      0001      000727 189L      0001      000756 190L      0001      000774 191L      0001      000751 192L
    C000      000226 2000F     0001      000113 201G      0001      000332 231L      0001      000343 232L      0001      000357 239L
    C001      001125 299L      0001      001145 304L      0001      000430 3066      0001      001212 309L      0001      001212 399L
    C001      000044 50L       0001      000276 531L      0000      000157 555F      0001      000675 611L      0000      000213 647F
    C000      000177 672F      0003  R   000020 ALINK      0000  R   000127 ALOW      0000  R   000130 ALPHA     0004      000064 AMM
    C004      000075 AMOFAC    0004      000077 AMOMAX    0004      000076 AMOMIN    0000  R   000077 AMONTE    0004      000100 AMORAT
    C004  R   000054 AMQ       0006      000123 AREA      0004      000107 BETTER    0004      000102 BIAS      0000  R   000102 BMQ
    C000  I   000001 BUFCNT    0006  I   000115 BUFPIX     0000  I   000000 BUFSIZ    0006  I   000116 BUFTOT    0003  R   000027 CIN
    C006  I   000002 CLSNAM    0000  R   000135 COFI      0005      000001 CONLV     0004      000111 CORLEN    0000  R   000040 COVEC
    C003  R   000030 CTOT      0003  R   000041 DCON      0004      000053 DELT      0010  I   000000 DISC      0003  R   000043 DISS
    C013  P   000000 DOTSQ     0006      000113 DUMP      0004      000073 DWFAC     0004      000061 ELIMIH    0004      000052 EPS
    C003  P   000050 GEN       0004      000074 GPACTM    0003  R   000047 GREF      0006      000000 IBEGIN    0003  I   000023 IDADJ
    C000  I   000112 IDO       0000  I   000136 IHOLD     0000  I   000105 INADDR    0003  I   000024 INDEX     0004      000066 INDXVL
    C000      000256 INJP$     0005      000003 IPT       0006      000130 IQ1       0000  I   000114 ISEC      0000  I   000107 ISTAT
    C000  I   000103 ITER      0003  I   000000 JUNK      0000  I   000123 KADJ      0000  I   000115 KPATH     0000  I   000113 KL
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0006    000107 KLBC       0000 I 000122 KPR        0004    000051 KROOT      0000 I 000101 KROT       0006 I 000127 LBUF
0006    000121 LBUFD      0003 I 000020 LINK       0004    000003 LKURT      0006    000102 LNCAT     0004    000007 LOSUN
0004    000004 LOVAR      0004 I 000002 LR         0004    000006 LSKEW      0003 I 000021 LSUBS      0004    000005 LSUM
0003 I  000022 LSUPER     0004 I 000005 LV         0004    000002 LVRIN      0005 R 000012 MACCEL     0006    000122 MAXBF
0004    000001 MM         0004 I 000110 MODE       0000 I  000076 MONTE      0004 I 000010 MO         0004 I  000011 MXAR
0003    000014 NARL       0000 I 000104 NBUFS      0006    000117 NBUFSD     0000 I 000111 NDO        0006    000120 NDUMP
0004    000010 NINCLS     0006 I 000106 NIT        0006    000004 NOFLD      0000 I 000110 NPIXEL     0006    000126 NPTS
0004 I  000070 NPTSO      0003 I 000043 NSYMB      0003 I  000020 NTB        0003    000016 NTBSZM    0003    000015 NTOP
0003    000017 NWANT      0006    000125 NWORS     0006    000124 NWDS       0000 I 000106 NWORDS     0007    000000 NX
0007 I  000001 NXA        0007 I 000002 NXO        0003 R  000045 OCIN       0004    000065 OCOCN     0003    000047 ODEN
0003 R  000046 OPRIOR     0003 R 000035 OPROP      0003 R  000036 OW         0000 R 000125 P          0005 R  000010 PACCEL
0003 R  000050 PCOND      0003 R 000025 PCUM       0004    000103 PJOIN      0000 R 000100 PLIM       0003 R  000043 PPASS
0000 R  000124 PPASSK     0003 R 000042 PRAT       0004    000071 PRATH      0003 R 000026 PRIRCM     0006 I  000103 PRNT
0003 R  000031 PROP       0000 R 000134 PROPL      0006 I  000111 PROUT      0006 I 000110 PRTME      0003 R  000044 PST
0000 R  000002 REL        0004    000065 SBLTH     0006 I  000114 SCRAM1     0004    000062 SEPTH     0005    000002 SKBND
0005    000003 SKCHI      0004 R 000112 SPCOR      0003 R  000032 SPFAC      0004    000072 SPMVTH    0000 R  000120 SPUSE
0006 I  000005 SYM        0006 I 000112 TOTPIX     0006 I  000001 TOTWRD     0005    000004 TRBND     0005    000005 TRCHI
0005    000006 URKBND     0005    000007 URKCHI    0005 R  000014 VACCEL     0004    000063 VFAC      0003 R  000037 VOLIN
0004    000101 VOLLIM     0003 R 000040 VOLRT      0003 R  000050 VRIN       0004    000104 VRJOIN    0003 R  000034 W
0003 R  000033 WADJ       0004    000060 WADJIN    0005    000000 WAIT       0004    000106 WDELSM    0000 R  000117 WDISS
0004    000067 WFAC       0000 R 000126 WO         0004    000105 WSIM       0004    000050 WTINIT    0000 R  000116 WUSE
0004 R  000056 XOVFLO     0004 R 000057 XUNFLO     0000 R  000133 ZFAC       0000 R 000131 ZO         0000 R  000132 ZOS
0000 R  000121 ZZ
```

```
00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101        1*          SUBROUTINE STATIS(KROTIN,PV,SUM,SKEW,KURT,OSUM,OVAR)                          000002
00101        2*      C   THIS PROGRAM TAKES EACH INPUT POINT AND CLASSIFIES IT                         000002
00101        3*      C                                                                    CH           000002
00101        4*      C  PURPOSE                                                           CH           000002
00101        5*      C   (1)  TAKE EACH INPUT POINT AND CLASSIFY IT (ON A FRACTIONAL,     CH           000002
00101        6*      C          PROBABILISTIC BASIS.)                                     CH           000002
00101        7*      C   (2)  UPDATES THE VARIOUS STATISTICAL PARAMETERS ASSOCIATED WITH THE CH         000002
00101        8*      C          CLASSES INDICATED.                                        CH           000002
00101        9*      C   (3)  CALLS ADJUST TO CHECK TO SEE IF ANY OF THESE CLASSES ARE     CH           000002
00101       10*      C          POTENTIALLY TWO AND REFER THOSE TO THE ROUTINE 'SPLIT'     CH           000002
00101       11*      C                                                                    CH           000002
00101       12*              INTEGER   BUFSIZ , BUFCNT                                                000002
00101       13*      C        (ON A FRACTIONAL, PROBABILISTIC BASIS).  IT THEN                        000002
00101       14*      C       UPDATES THE VARIOUS STATISTICAL PARAMETERS ASSOCIATED                    000002
00101       15*      C        WITH THE CLASSES INDICATED AND CHECKS TO SEE IF                         000002
00101       16*      C        ANY OF THESE CLASSES IS POTENTIALLY TWO.  THOSE WHICH                   000002
00101       17*      C        ARE ARE REFERRED TO THE ROUTINE 'SPLIT'.                                000002
00104       18*              INCLUDE MISH                                                            000002
00104       19*      C                                                                               000002
00132       20*              INCLUDE CMBK10                                                          000002
00141       21*              COMMON /RAND/NX,NXA,NXO                                                 000002
00141       22*      CHANGE***     1                                                                 000002
00142       23*              REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)                             000002
00142       24*      C                                                                               000002
00141       25*              DIMENSION   REL(30) , COVEC(30) ,  PV(MQ,BUFPIX)                        000002
00143       26*      C                                                                               000002
00144       27*              DATA MONTE,AMONTE,PLIM/3,3,.1/                                          000002
00148       28*      C  MONTE---CHECK LIK. RATIO 1/3 OF THE TIME, ELIM. WHEN PROP. LT /.1     CH     000002
00150       29*              INTEGER DISC                                                            000002
```

```
      XP(DIST)=EXP(-.5*DIST)
C     HERE ABOVE GDET IS THE SQUARE ROOT OF THE COVARIANCE
C          MATRIX, AND FFAC IS A POWER OF PI.
      KROT=KROTIN
C
      BMQ=.6666666667*AMQ
C
C
C                                                          CH
C              ***** READ AND PROCESS DATA  NIT  TIMES *****
C
      ITER = 0
C
    1 ITER = ITER + 1
C
C                  *** READ 1 BUFFER OF SCRAMBLED DATA ***
C
C  MQ = LENGTH OF ONE VECTOR                               CH
      BUFSIZ = BUFPIX * MQ
      NBUFS = TOTWRD/BUFSIZ
      LBUF = MOD( TOTWRD , BUFSIZ )
      IF( LBUF .GT. 0 )   NBUFS = NBUFS + 1
C  INADDR = 1ST WORD OF ORIG DATA ON FAST STORAGE          CH
      INADDR = SCRAM1
      BUFCNT = 0
C  TOTWRD = NO. WORDS IN ORIGINAL DATA ON DRUM             CH
C  MQ = LENGTH OF ONE VECTOR                               CH
      TOTPIX = TOTWRD/MQ
C
   50 BUFCNT = BUFCNT + 1
C
      NWORDS = BUFSIZ
      IF ( LBUF .GT. 0 .AND. BUFCNT .EQ. NBUFS )  NWORDS = LBUF
C
C
      CALL RREAD (INADDR, PV, NWORDS, ISTAT)
   10 IF( ISTAT .GT. 0 )   GO TO 10
C
      INADDR = INADDR + NWORDS
      NPIXEL = NWORDS/MQ
      NDO = NPIXEL
C
C
C       *** INSPECT EACH CLASS AND PROCESS EACH OF THE DATA POINTS  ***
      DO 399 IDO=1,NDO
C  ** THIS CODE GETS RANDOM NUMBERS. **
C  GET NEXT POINT IN SEQUENCE
C                                                          CH
C                                                          CH
C     WE USE MONTE-CARLO TECHNIQUES FOR LOW PROBABILITY CLASSES(P<PLIM)
      PCUM(KROT)=0.
C
      IF( INDEX(KL) .NE. 0 .AND. KL .NE. 119)
     *   WRITE(6,1000)    IDO , INDEX(KL) , KL
 1000 FORMAT(    3X, '**** WARNING FROM STATIS ****  ON THE ', 2X, I5,
     * 2X, 'TIME, INDEX(KL)=' , I5, 3X  ', KL=' , I5     )
C
```

```
00213         PRIRCM(KROT)=0.                                                      000145
00214         PPASS(KROT)=1.                                                       000147
00215         ISEC=0                                                               000151
00216         KL=LSUBS(KROT)                                                       000152
00217         KFATH=KROT                                                           000154
00217   C   GO DOWN CLUSTER TREE                                                   000154
00220     130 IF(LSUBS(KL).EQ.0) GO TO 131                                         000157
00220   C                                                             CH           000157
00220   C   FIND BOTTOM NODE                                          CH           000157
00222         PCUM(KL)=0.                                                          000161
00223         PRIRCM(KL)=0.                                                        000162
00224         KFATH=KL                                                             000163
00225         KL=LSUBS(KL)                                                         000165
00226         GO TO 130                                                            000167
00226   CHANGE***     4                                                            000167
00226   C                                                             CH           000167
00226   C   CALC UNWEIGHTED NORMALIZED VECTOR REL                     CH           000167
00226   C     WUSE = CURRENT WEIGHT                                   CH           000167
00227     131 IF(INDEX(KL).LE.0) GO TO 133                                         000171
00227   C                                                             CH           000171
00227   C         USE NEW WEIGHTS AND MEANS IF ADJUST HAS BEEN CALLED CH           000171
00231         CALL COPECT(REL,PV(1,IDO),W(KL),SUM(KL+1))                           000177
00232         WUSE=W(KL)                                                           000217
00233         PROP(KL)=CIN(KL)/(W(KFATH)-CTOT(KL))                                 000221
00234         GO TO 134                                                            000226
00234   CHANGE***     4.5                                                          000226
00235     133 CALL COPECT(REL,PV(1,IDO),OW(KL),OSUM(KL+1))                         000230
00236         WUSE=OW(KL)                                                          000252
00237     134 DISS(KL)=DOTSQ(REL,WRIN(KL+1))*WUSE                                  000255
00240         WDISS = DISS(KL) + DCON(KL)                                          000265
00241         IF(ABS(WDISS).LE. 100.) GO TO 531                                    000267
00243         PCOND(KL)=0.                                                         000273
00244         GO TO 138                                                            000274
00245     531 CONTINUE                                                             000276
00246         PCOND(KL)=XP(WDISS)/VOLRT(KL)                                        000276
00247     138 IF(LSUBS(KL .NE.0) PCUM(KL)=PCUM(KL)/PRIRCM(KL)                      000307
00251         SPUSE=SPFAC(KL)/SPCOR                                                000316
00252         IF(SPUSE.GT.XUNFLO) GO TO 231                                        000322
00254         PST(KL)=PROP(KL)*PCOND(KL)                                           000325
00254   C                                                                          000325
00254   C   SET KL = LAST NODE IN STRING                                           000325
00255         GO TO 239                                                            000330
00256     231 IF(SPUSE.LT.XOVFLO) GO TO 232                                        000332
00260         PST(KL)=PPOP(KL)*PCUM(KL)                                            000336
00261         GO TO 239                                                            000341
00262     232 ZZ=EXP(SPUSE)                                                        000343
00263         PST(KL)=PROP(KL)*(PCOND(KL)+ZZ*PCUM(KL))/(1.+ZZ)                     000347
00264     239 PCUM(KFATH)=PCUM(KFATH)+PST(KL)                                      000357
00265         PRIRCM(KFATH)=PRIRCM(KFATH)+PROP(KL)                                 000363
00266     139 KL=LINK(KL)                                                          000366
00267         IF(KL)130,149,130                                                    000370
00267   C   GO UP TREE                                                             000370
00272     149 KL=KFATH                                                             000371
00273         KFATH=LSUPER(KL)                                                     000374
00274         IF(KL.NE.KROT) GO TO 131                                             000376
00274   C   WE NOW HAVE THE RELEVANT CLASSES AND THEIR PROBABILITIES AVAILABLE.    000376
00274   C                                                                          000376
```

OF POOR QUALITY

```
00274  144*      C  NEXT WE MAKE THE APPROPRIATE INDIVIDUAL FIRST-ORDER STATISTICS ADJ.          000376
00276  145*        150 CONTINUE                                                                 000400
00277  146*            PCUM(KROT)=PCUM(KROT)/PRIRCM(KROT)                                       000401
00300  147*            IF(PCUM(KROT).NE.0.) GO TO 151                                           000404
       148*      CHANGE***      5
00302  149*            PRINT 555,IDO,W(KROT),(PV(KPR,IDO),KPR=1,MQ)                             000404
00312  150*        555 FORMAT('0**SUSPECTED BAD DATA POINT--STATIS**IDO=',I5,'   ROOT',         000406
00312  151*           1   F10.2/5X,'VECTOR',(5E12.6))                                         000433
00313  152*            GO TO 399                                                                000433
00314  153*        151 CONTINUE                                                                 000435
00315  154*            KL=LSUBS(KROT)                                                           000435
00316  155*            KFATH=KROT                                                               000437
00317  156*            W(KROT)=W(KROT)+PPASS(KROT)                                              000441
00320  157*            NPTSO=NPTSO+1                                                            000444
00321  158*            KADJ=0                                                                   000447
00322  159*        153 CONTINUE                                                                 000451
00323  160*            IF(PST(KL).EQ.0.) GO TO 299                                              000451
00325  161*            PPASSK=PPASS(KFATH)                                                      000453
00326  162*            P=PST(KL)/(PCUM(KFATH)*PRIRCM(KFATH))*PPASSK                             000456
00326  163*      C       KLO=KL                                                                 000456
00327  164*        132 IF(P.GE.PLIM)GO TO 140                                                   000465
00331  165*            IF(DISC(MONTE).NE.1)GO TO 299                                            000470
00331  166*            PPASSK=PPASSK*AMONTE                                                     000475
00334  167*            P=P*AMONTE                                                               000500
00335  168*            GO TO 132                                                                000503
       169*      CHANGE***      6
00336  170*        140 IF(INDEX(KL).LE.0) GO TO 143                                           000503
00340  171*            CALL CORECT(REL,PV(1,IDO),W(KL),SUM(KL+1))                               000505
00341  172*            GO TO 144                                                                000510
00341  173*      CHANGE***      6.5
00342  174*        143 CALL CORECT(REL,PV(1,IDO),OW(KL),OSUM(KL+1))                           000532
00341  175*        144 WO=W(KL)                                                               000532
00344  176*            IF(P.GT.1.001.OR.P.LT.0.)PRINT 672,INDEX(KL),KL,INDEX(KFATH),          000534
00344  177*           1  KFATH,IDO,P,PST(KL),PCUM(KFATH),PRIRCM(KFATH),PPASSK,               000557
00345  178*           2  PROP(KL)                                                            000561
00362  179*        672 FORMAT(' PROB ERROR(STATIS):',2(I3,I7),I6,' P=',E9.4,                  000561
00362  180*           1   20X,'FROM',7E9.4)                                                  000561
00363  181*            IF (P.GT.1.1) P=.01                                                    000624
00365  182*            W(KL)=W(KL)+P                                                          000624
00366  183*            ALOW=P/W(KL)                                                           000624
00367  184*            ALPHA=WO*ALOW                                                          000632
00367  185*      C   HERE WE ADJUST SPFAC AND PQRAT.                                          000636
00370  186*            IF(LSUBS(KL).EQ.0) GO TO 611                                           000641
00372  187*            ZQ=(PCUM(KL)-PCOND(KL))/(PCUM(KL)+PCOND(KL)+1.E-37)                    000643
00373  188*            ZQS=ZQ*ZQ                                                              000645
00374  189*            PQRAT(KL)=PQRAT(KL)+P*ZQS                                              000654
00375  190*            SPFAC(KL)=SPFAC(KL)+P*ZQ*(2.+ZQS/(1.5-.9*ZQS))                         000656
00376  191*        611 CONTINUE                                                               000661
00377  192*            IF(INDEX(KL).LT.0) GO TO 189                                           000675
00401  193*            VOLIN(KL)=VOLIN(KL)*(1.+ALOW*DISS(KL))                                 000675
00402  194*            VOLRT(KL)=.5*(VOLRT(KL)+VOLIN(KL)/VOLRT(KL))                          000700
00402  195*      C       HERE WE KEEP VOLRT NEAR SQRT(VOLIN) BY NEWTON'S METHOD.              000705
00403  196*            ZFAC=.5/(1./ALOW+.5)                                                   000705
00404  197*            DCON(KL)=DCON(KL)-BMQ*ZFAC*(3.+ZFAC*ZFAC)                              000711
00404  198*      C     THE ABOVE APPROXIMATION TO THE LOG IS CORRECTED FOR IN ADJUST          000717
00404  199*      C   PROPORTION CALCULATION.                                                  000717
00405  200*        189 PROPL=PROP(KL)/PRIRCM(KFATH)                                         000717
                                                                                            000727
```

```
      IF(PROPL.GT..99999) GO TO 192
      IF(P.GE,PPASSK*PROPL) GO TO 190
      CTOT(KL)=CTOT(KL)+P/PROPL
      GO TO 191
  192 CIN(KL)=CIN(KL)+1.
      GO TO 191
  190 CTOT(KL)=CTOT(KL)+(PPASSK-P)/(1.-PROPL)
      CIN(KL)=CIN(KL)+(P-PPASSK*PROPL)/(1.-PROPL)
  191 CONTINUE
CHANGE***     7
      CALL VPV(SUM(KL+1),P,PV(1,IDO))
      IF(INDEX(KL).LE.0) GO TO 163
      CALL VMTV(COVEC,VRIN(KL+1),REL)
      COFI = -ALPHA/(1.+ ALOW*DISS(KL))
      CALL MPVS(VRIN(KL+1),COFI,COVEC)
      GO TO 164
  163 CALL MPVS(OVAR(KL+1),ALPHA,REL)
  164 CONTINUE
C  DISS(KL) CONTAINS THE GAUSSIAN DISTANCE OF THE POINT FROM THE CLUSTER
C  VRIN IS THE INVERSE COVARIANCE MATRIX    (*****) OVER W(KL)     (**** NOTE)
C                       (THIS INTRODUCES SEVERAL SCALE FACTORS )
C  COV.C IS THE CONTRAVARIANT FORM OF THE RELATIVE DISTANCE REL,
C             COVEC=VRIN*REL
C  WE NOW HAVE ALL THE LINEAR AND QUADRATIC STATISTICS,AND PROCEED
C      TO CALCULATE THE APPROXIMATE 3RD AND 4TH MOMENTS FOR TESTING.
C    THESE MOMENTS ARE NOT CALCULATED EXACTLY: THE SQUARED
C          DISTANCE OF A POINT FROM THE MEAN ACTUALLY SHOULD
C          USE ALL THE DATA IN CALCULATING THE MEAN AND
C          COVARIANCE, WHEREAS WE SUBSTITUTE THE CURRENT
C          VALUES INSTEAD.  THUS THE VALUES CALCULATED DEPEND
C          ON THE ORDER THE POINTS ARE READ IN.  THIS IS NOT
C            CRITICAL.
C
C  SKIP CALC IF THERE ARE SUBS FOR THIS CLUSTER
C
      WDISS=DISS(KL)*P
      CALL VPV(SKEW(KL+1),WDISS,REL)
      CALL MPVS(KURT(KL+1),WDISS,REL)
298   CONTINUE
C
C  WE NOW ADJUST THE CLASS FOR LARGE-SCALE STATISTICAL EFFECTS,
C     ON AN OCCASIONAL BASIS.    THIS INCLUDES NOMINAL NEWTONS
C      METHOD CORRECTIONS AND TESTING FOR THE POSSIBILITY
C     OF TWO CLUSTERS (USING THE SKEW AND KURT STATISTICS).
      IF(W(KL).GT.WADJ(KL).OR.NPTSO.GE.IDADJ(KL)) KADJ=KL
  299 PPASS(KL)=P
      IF(LSUBS(KL).EQ.0.OR.PCUM(KL).EQ.0.) GO TO 304
      KFATH=KL
      KL=LSUBS(KL)
      GO TO 153
  304 KL=LINK(KL)
  303 IF(KL) 153,305,153
  305 KL=KFATH
      KFATH=LSUPER(KL)
      IF(KL.NE.KROT) GO TO 304
      IF(KADJ.NE.0) CALL ADJUST(KADJ,SUM,SKEW,KURT,OSUM,OVAR)
      IF(MOD(NPTSO,TOTPIX).NE.0.OR.MODE.EQ.0) GO TO 309
```

```
00462           NXA=NXO                                                        001204
00463           CALL CLDUMP(KROT)                                              001206
00464       309 CONTINUE                                                       001213
00465       647 FORMAT('0LOOP IN STATIS:IDO,W(KROT),KL,SECTION',I5,E11.5,2I5  001213
00465         1     /(1X14I5))                                                 001213
00466       399 CONTINUE                                                       001213
00466     CHANGE***      8                                                     001213
00470           IF( BUFCNT .LT. NBUFS )  GO TO 50                              001213
00472           IHOLD = PROUT                                               NEW001217
00473           WRITE  (6,2000) ITER                                        NEW001221
00476      2000 FORMAT('      ITERATIONS THROUGH ALL THE DATA = ',I4)       NEW001227
00477           IF ( ITER .LT. NIT)  GO TO 1                                   001227
00501           RETURN                                                         001233
00502           END                                                            001304
```

END OF COMPILATION: 1 DIAGNOSTICS.

```
@FOR,S STOFLO,STOFLO,STOFLO
FOR SOE3=04/18/78-01:35:01 (0,0)

   SUBROUTINE STOFLO    ENTRY POINT 000017

   STORAGE USED: CODE(1) 000021; DATA(0) 000024; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    NPRT$
    0007    NIO2$
    0010    NSTOP$
    0011    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000000 101F      0003 R 000020 ALINK     0004   000064 AMM      0004   000075 AMOFAC   0004   000077 AMOMAX
    0004    000076 AMOMIN    0004   000100 AMORAT    0004   000054 AMQ      0004   000107 BETTER   0004   000102 BIAS
    0003 R  000027 CIN       0005   000001 CONLV     0004   000111 CORLEN   0003 R 000030 CTOT     0003 R 000041 DCON
    0004    000053 DELT      0003 R 000043 DISS      0004   000073 DWFAC    0004   000061 ELIMTH   0004   000052 EPS
    0003 R  000050 GEN       0004   000074 GRACTM    0003 R 000047 GREF     0003 I 000023 IDADJ    0003 I 000024 INDEX
    0004    000066 INDXVL    0000   000020 INJPS     0003 I 000000 JUNK     0004   000051 KROOT    0003 I 000020 LINK
    0004    000003 LKURT     0004   000007 LOSUM     0004   000004 LOVAR    0004 I 000002 LR       0004   000006 LSKEW
    0003 I  000021 LSUBS     0004   000005 LSUM      0003 I 000022 LSUPER   0004 I 000005 LV       0004   000002 LVRIN
    0005 R  000012 MACCEL    0004   000001 MM        0004   000110 MODE     0004   000000 MQ       0004 I 000011 MXAR
    0003 I  000014 NARL      0004   000010 NINCLS    0004   000070 NPTS0    0003 I 000043 NSYMB    0003 I 000020 NTB
    0003    000016 NTBSZM    0003 I 000015 NTOP      0003   000017 NWANT    0003 R 000045 OCIN     0004   000055 ODCON
    0003 R  000047 ODEN      0003 R 000046 OPRIOR    0003 R 000035 OPROP    0003 R 000036 OW       0005 R 000010 PACCEL
    0003 R  000050 PCOND     0003 R 000025 PCUM      0004   000103 PJGIN    0003 R 000043 PPASS    0003 R 000042 PQRAT
    0004    000071 PQRATH    0003 R 000026 PRIRCM    0003 R 000031 PPOP     0003 R 000044 PST      0004   000065 SBLTH
    0004    000062 SEPTH     0005   000002 SKBND     0005   000003 SKCHI    0004   000112 SPCOR    0003 R 000032 SPFAC
    0004    000072 SPMVTH    0005   000004 TPBND     0005   000005 TRCHI    0005   000006 URKBND   0005   000007 URKCHI
    0005 R  000014 VACCEL    0004   000063 VFAC      0003 R 000037 VOLIN    0004   000101 VOLLIM   0003 R 000040 VOLRT
    0003 R  000050 VRIN      0004   000104 VRJOIN    0003 R 000034 W        0003 R 000033 WADJ     0004   000060 WADJIN
    0005    000000 WAIT      0004   000106 WDELSM    0004   000067 WFAC     0004   000105 WSIM     0004   000050 WTINIT
    0004    000056 XOVFLO    0004   000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            SUBROUTINE STOFLO                                                        000000
00101      2*      C      IF A RETURN IS MADE TO CALLOC FROM THIS ROUTINE, THE                    000000
00101      3*      C          'FREE' ROUTINE WILL TRY TO REALLOCATE, AND WILL CALL                000000
00101      4*      C          'STOFLO' AGAIN IF NTOP AND NARL ARE UNSATISFACTORY.                 000000
00101      5*            INCLUDE MISH                                                             000000
00131      6*            PRINT 101,NTOP,NARL                                                      000000
```

```
00134       7*        101 FORMAT('1STORAGE OVERFLOW ERROR. NEXT FREE=',I6,                    000005
00135       8*           1    ' LIMIT=',I6,  '     EXECUTION TERMINATED.')                  000005
00136       9*             STOP                                                              000005
00137      10*             END                                                               000020

      END OF COMPILATION:       1 DIAGNOSTICS.
```

```
@FOR,S SUBLIM,SUBLIM,SUBLIM
FOR SOE3,-04/18/78-01:35:04 (0,0)

   SUBROUTINE SUBLIM    ENTRY POINT 000054

   STORAGE USED: CODE(1) 000063; DATA(0) 000027; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    TRFREE
    C007    NPRT$
    C010    NIO2$
    C011    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000015 11L      0000    000003 713F     0000    000015 714F     0003 R 000020 ALINK    0004    000064 AMM
    C004    000075 AMOFAC   0004    000077 AMOMAX   0004    000076 AMOMIN   0004    000100 AMORAT   0004    000054 AMQ
    C004    000107 BETTER   0004    000102 BIAS     0003 R  000027 CIN      0005    000001 CONLV    0004    000111 CORLEN
    C003 R  000030 CTOT     0003 R  000041 DCON     0004    000053 DELT     0003 R  000043 DISS     0004    000073 DWFAC
    C004    000061 ELIMTH   0004    000052 EPS      0003 R  000050 GEN      0004    000074 GRACTM   0003 R  000047 GREF
    C003 I  000023 IDADJ    0003 I  000024 INDEX    0004    000066 INDXVL   0000    000017 INJPS    0003 I  000000 JUNK
    C000 I  000001 K        0000 I  000000 KL       0000 I  000002 KNX      0004    000051 KROOT    0003 I  000020 LINK
    C004    000003 LKURT    0004    000007 LOSUM    0004    000004 LOVAR    0004 I  000002 LR       0004    000006 LSKEW
    C003 I  000021 LSUBS    0004    000005 LSUM     0003 I  000022 LSUPER   0004 I  000005 LV       0004    000002 LVRIN
    C005 R  000012 MACCEL   0004    000001 MM       0004    000110 MODE     0004    000000 MQ       0004 I  000011 MXAR
    C003    000014 NARL     0004 I  000010 NINCLS   0004    000070 NPTSD    0003 I  000043 PSYMB    0003 I  000020 NTB
    C003    000016 NTBSZM   0003    000015 NTOP     0003    000017 NWANT    0003 R  000045 OCIN     0004    000055 ODCON
    C003 P  000047 ODEN     0003 R  000046 OPRIOR   0003 R  000035 OPROP    0003 R  000036 OW       0005 R  000010 PACCEL
    C003 P  000050 PCOND    0003 R  000025 PCUM     0004    000103 PJOIN    0003 R  000043 PPASS    0003 R  000042 PQRAT
    C004    000071 PQRATH   0003 R  000026 PRIRCM   0003 R  000031 PROP     0003 R  000044 PST      0004 R  000065 SBLTH
    C004    000062 SEPTH    0005    000002 SKBND    0005    000003 SKCHI    0004    000112 SPCOR    0003 R  000032 SPFAC
    C004    000072 SPMVTH   0005    000004 TRBND    0005    000005 TRCHI    0005    000006 URKBND   0005    000007 URKCHI
    C005 R  000014 VACCEL   0004    000063 VFAC     0003 R  000037 VOLIN    0004    000101 VOLLIM   0003 R  000040 VOLRT
    C003 R  000050 VRIN     0004    000104 VPJOIN   0003 R  000034 W        0003 R  000033 WADJ     0004    000060 WADJIN
    C005    000000 WAIT     0004    000106 WDELSM   0004    000067 WFAC     0004    000105 WSIM     0004    000050 WTINIT

000000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
000101      1*            SUBROUTINE SUBLIM(KLHED)
000101      2*        C   SUBLIM ELIMINATES THE SUBCLUSTERS OF THE NODE KLHED.                    000000
000101      3*            INCLUDE MISH                                                            000000
000131      4*            PRINT 713,INDEX(KLHED),SPFAC(KLHED),SBLTH                               000000
000136      5*          713 FORMAT ('0###SUB ELIM  ',I3,'   SPLITTING',E12.5,'  <',E12.5)         000007
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00137      6*            KL=KLHED                                          000007
0014C      7*            K=LSUBS(KL)                                       000012
00141      8*         11 KNX=LINK(K)                                       000015
00142      9*            PRINT 714 ,INDEX(K)                               000017
00145     10*            CALL TRFREE(K,NINCLS)                             000025
00146     11*        714 FORMAT(I15)                                       000031
00147     12*            K=KNX                                             000031
0015C     13*            IF(K.NE.0) GO TO 11                               000033
00152     14*            LSUBS(KL)=0                                       000035
00151     15*            SPFAC(KL)=-9999.                                  000036
00154     16*            PORAT(KL)=0.                                      000040
00155     17*            RETURN                                            000041
0015E     18*            END                                               000062

      END OF COMPILATION:          1 DIAGNOSTICS.
```

```
@FOR,S TR,TR,TR
FOR SDE3-04/18/78-01:35:08 (0,0)

   FUNCTION TR          ENTRY POINT 000105

   STORAGE USED: CODE(1) 000124; DATA(3) 000035; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003   CLUS   002017
    C004   MISC   000113
    C005   STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006   NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000035 134G      0001    000046 142G      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    C004    000077 AMOMAX    0004    000076 AMOMIN    0004   000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER
    C004    000102 BIAS      0003 R  000027 CIN       0005   000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT
    C003 R  000041 DCON      0004    000053 DELT      0003 R 000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH
    C004    000052 EPS       0003 R  000050 GEN       0004   000074 GRACTM    0003 R  000047 GREF      0000 I  000001 I
    C003 I  000023 IDADJ     0003 I  000024 INDEX     0004   000066 INDXVL    0000    000005 INJPS     0003 I  000000 JUNK
    C004    000051 KROOT     0003 I  000020 LINK      0004   000003 LKURT     0004    000007 LOSUM     0004    000004 LOVAR
    C004 I  000002 LR        0004    000006 LSKEW     0003 I 000021 LSUBS     0004    000005 LSUM      0003 I  000022 LSUPER
    C004 I  000005 LV        0004    000002 LVRIN     0005 R 000012 MACCEL    0004 I  000001 MM        0004    000110 MODE
    C004 I  000000 MQ        0000 I  000002 MXA       0004 I 000011 MXAR      0003    000014 NARL      0004    000010 NINCLS
    C004    000070 NPTSO     0003 I  000043 NSYMB     0003 I 000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP
    C003    000017 NWANT     0003 R  000045 OCIN      0004   000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR
    C003 R  000035 OPROP     0003 R  000036 OW        0005 R 000010 PACCEL    0003 R  000050 PCOND     0003 R  000025 PCUM
    C004    000103 PJOIN     0003 R  000043 PPASS     0003 R 000042 PQRAT     0004    000071 PQRATH    0003 R  000026 PRIRCM
    C003 R  000031 PROP      0003 R  000044 PST       0004   000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND
    C005    000003 SKCHI     0004    000112 SPCOR     0003 R 000032 SPFAC     0004    000072 SPMVTH    0000 R  000000 TR
    C005    000004 TRBND     0005    000005 TRCHI     0005   000006 URKBND    0005    000007 URKCHI    0005 R  000014 VACCEL
    C004    000063 VFAC      0003 R  000037 VOLIN     0004   000101 VOLLIM    0003 R  000040 VOLRT     0003 R  000050 VRIN
    C004    000104 VRJOIN    0003 R  000034 W         0003 R 000033 WADJ      0004    000060 WADJIN    0005    000000 WAIT
    C004    000106 WDELSM    0004    000067 WFAC      0004   000105 WSIM      0004    000059 WTINIT    0004    000056 XOVFLO
    C004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*          FUNCTION TR(AM,AMET)                                                    000023
00101       2*      C     CALCULATES THE TRACE OF THE MATRIX AM RELATIVE TO THE METRIC AMET.     000023
00101       3*            INCLUDE MISH                                                          000023
00131       4*            REAL AM(475),AMET(475)                                                000023
00132       5*            TR=  AM(1)*AMET(1)*.5                                                 000023
00133       6*            DO 10 I=2,MM                                                          000035
00136       7*         10 TR=TR+AM(I)*AMET(I)                                                   000035
00140(      8*            TR=TR+TR                                                              000041
```

```
0014 C      9*      C       WE MUST DOUBLE THE OFFDIAGONAL TERMS (SEE COMMENT IN FUNCTION DOT)      000041
0014 C     10*      C       NOW SUBTRACT DIAGONALS.                                                 000041
0014 1     11*            DO 15 I=2,MD                                                             000046
0014 4     12*            MXA=MXAR(I)                                                              000046
0014 5     13*         15 TR=TR-AM(MXA+I)*AMET(MXA+I)                                              000055
0014 7     14*            RETURN                                                                   000063
0015 C     15*            END                                                                      000123

      END OF COMPILATION:      1 DIAGNOSTICS.
```

```
@FOR,S TRFREE,TRFREE,TRFREE
FOR SOE3-04/18/78-01:35:12 (0,0)

   SUBROUTINE TRFREE    ENTRY POINT 000052

   STORAGE USED: CODE(1) 000064; DATA(0) 000011; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    FREE
    C007    NERR3$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001      000017 11L          0001    000007 9L           0001    000041 99L          0003 R 000020 ALINK        0004    000064 AMH
    C004      000075 AMOFAC       0004    000077 AMOMAX       0004    000076 AMOMIN       0004   000100 AMORAT       0004    000054 AMS
    C004      000107 BETTER       0004    000102 BIAS         0003 R  000027 CIN          0005   000001 CONLV        0004    000111 CORLEN
    C003  P   000030 CTOT         0003 R  000041 DCGN         0004    000053 DELT         0003 R 000043 DISS         0004    000073 DWFAC
    C004      000061 ELIMTH       0004    000052 EPS          0003 R  000050 GEN          0004   000074 GRACTM       0003 R  000047 GREF
    C003  I   000023 IDADJ        0003 I  000024 INDEX        0004    000066 INDXVL       0000   000004 INJPS        0003 I  000000 JUNK
    C000  I   000000 KL           0000 I  000002 KLF          0000 I  000003 KLK          0000 I 000001 KLO          0004    000051 KROOT
    C003  I   000020 LINK         0004    000003 LKURT        0004    000007 LOSUM        0004   000008 LOVAR        0004 I  000002 LR
    C004      000006 LSKEW        0003 I  000021 LSUBS        0004    000005 LSUM         0003 I 000022 LSUPER       0004 I  000005 LV
    C004      000002 LVRIN        0005 R  000012 MACCEL       0004    000001 MM           0004   000110 MODE         0004    000000 MQ
    C004  I   000011 MXAR         0003    000014 NARL         0004    000010 NINCLS       0004   000070 NPTSO        0003 I  000043 NSYMB
    C003  I   000020 NTB          0003    000016 NTBSZM       0003    000015 NTOP         0003   000017 NWANT        0003 R  000045 OCIN
    C004      000055 ODCON        0003 R  000047 ODEN         0003 R  000046 OPRIOR       0003 R 000035 OPROP        0003 R  000036 OW
    C005  R   000010 PACCEL       0003 R  000050 PCOND        0003 R  000025 PCUM         0004   000103 PJOIN        0003 R  000043 PPASS
    C003  R   000042 PQRAT        0004    000071 PQRATH       0003 R  000026 PRIRCM       0003 R 000031 PROP         0003 R  000044 PST
    C004      000065 SBLTH        0004    000062 SEPTH        0005    000002 SKBND        0005   000003 SKCHI        0004    000112 SPCOR
    C003  R   000032 SPFAC        0004    000072 SPMVTH       0005    000004 TRBND        0005   000005 TRCHI        0005    000006 URKBND
    C005      000007 URKCHI       0005 R  000014 VACCEL       0004    000063 VFAC         0003 R 000037 VOLIN        0004    000101 VOLLIM
    C003  R   000040 VOLRT        0003 R  000050 VRIN         0004    000104 VPJOIN       0003 R 000034 W            0003 R  000033 WADJ
    C004      000060 WADJIN       0005    000000 WAIT         0004    000106 WDELSM       0004   000067 WFAC         0004    000105 WSIM
    C004      000050 WTINIT       0004    000056 XOVFLO       0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00001       1*          SUBROUTINE TRFREE(KLHED,LEN)                                                  000000
00001       2*      C  THIS ROUTINE FREES THE TREE HEADED BY KLHED.                                   000000
00001       3*      C   THE USER ROUTINE MUST INSURE THAT PINTERS TO KLHED,ETC., ARE                  000000
00001       4*      C      PROPERLY ADJUSTED.                                                         000000
00001       5*          INCLUDE MISH                                                                  000000
00013       6*          IF(KLHED.EQ.0) RETURN                                                         000000
00013       7*          KL=KLHED                                                                      000004
```

```
00134      8*          9 KLO=KL                                                000007
00135      9*            KL=LSUBS(KL)                                          000011
00136     10*            IF(KL.NE.0) GO TO 9                                   000013
00140     11*            KL=KLO                                                000015
00141     12*         11 KLF=LSUPER(KL)                                        000017
00142     13*         10 KLK=LINK(KL)                                          000021
00143     14*            CALL FREE(KL,LEN)                                     000023
00144     15*            IF(KL.EQ.KLHED) GO TO 99                              000027
00146     16*            KL=KLK                                                000032
00147     17*            IF(KL)9,13,9                                          000034
00152     18*         13 KL=KLF                                                000035
00153     19*            GO TO 11                                              000037
00154     20*         99 KLHED=0                                               000041
00155     21*            RETURN                                                000041
00156     22*            END                                                   000063

      END OF COMPILATION:      1 DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

BEST AVAILABLE COPY
OF POOR QUALITY

```
@FOR,S TRIMTX,TRIMTX,TRIMTX
FOR SOE3-04/18/78-01:35:15 (0,0)

   SUBROUTINE TRIMTX    ENTRY POINT 000056

   STORAGE USED: CODE(1) 000065; DATA(0) 000022; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS   002017
    0004    MISC   000113
    0005    STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000013 133G      0001    000030 140G      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC
    0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER
    0004    000102 BIAS      0003 R  000027 CIN       0005    000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT
    0003 R  000041 DCON      0004    000053 DELT      0003 R  000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH
    0004    000052 EPS       0003 R  000050 GEN       0004    000074 GRACTM    0003 R  000047 GREF      0000 I  000000 I
    0003 I  000023 IDADJ     0000 I  000002 IJ        0003 I  000024 INDEX     0004    000066 INDXVL    0000    000005 INJPS
    0000 I  000003 J         0003 I  000000 JUNK      0004    000051 KROOT     0003 I  000020 LINK      0004    000003 LKURT
    0004    000007 LOSUM     0004    000004 LOVAR     0004 I  000002 LR        0004    000006 LSKEW     0003 I  000021 LSUBS
    0004    000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV        0004    000002 LVRIN     0005 R  000012 MACCEL
    0004    000001 MM        0004    000110 MODE      0004 I  000000 MQ        0000 I  000001 MX        0004 I  000011 MXAR
    0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSO     0003 I  000043 NSYMB     0003 I  000020 NTB
    0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT     0003 R  000045 OCIN      0004    000055 ODCON
    0003 R  000047 ODEN      0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW        0005 R  000010 PACCEL
    0003 R  000050 PCOND     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS     0003 R  000042 PQRAT
    0004    000071 PQRATH    0003 R  000026 PRIRCH    0003 R  000031 PROP      0003 R  000044 PST       0004    000065 SBLTH
    0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR     0003 R  000032 SPFAC
    0004    000072 SPMVTH    0005    000004 TRBND     0005    000005 TRCHI     0005    000006 URKBND    0005    000007 URKCHI
    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN     0004    000101 VOLLIM    0003 R  000040 VOLRT
    0003 R  000050 VRIN      0004    000104 VRJOIN    0003 R  000034 W         0003 R  000033 WADJ      0004    000060 WADJIN
    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIM      0004    000050 WTINIT
    0004    000056 XOVFLO    0004    000057 XUNFLO

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*            SUBROUTINE TRIMTX(TRI,SQ)                                              000013
00101       2*      C   THIS ROUTINE TAKES THE LOWER TRIANGLE OF SQ(DIM MQ*MQ) AND PUTS           000013
00101       3*      C       IT INTO SYMMETRIC MATRIX FORM IN TRI.                                  000013
00101       4*            INCLUDE MISH                                                           000013
00131       5*            DIMENSION TRI(475),SQ(900)                                             000013
00132       6*            DO 10 I=1,MQ                                                           000016
00136       7*            MX=MXAR(I)                                                             000022
00136       8*            IJ=I
```

```
00137      9*          DO 10 J=1,I                                          000030
00140     10*          TRI(MX+J)=SQ(IJ)                                     000031
00143     11*    10    IJ=IJ+MQ                                             000033
00146     12*          RETURN                                               000041
00147     13*          END                                                  000064

      END OF COMPILATION:          1 DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@FOR,S VMTV,VMTV,VTMV
FOR SOE3-04/18/78-01:36:26 (0,0)

   SUBROUTINE VMTV      ENTRY POINT 000115

   STORAGE USED: CODE(1) 000132; DATA(0) 000032; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001    000016 134G       0001    000026 140G       0001    000057 152G       0001    000073 20L        0003 R  000020 ALINK
    C004    000064 AMM        0004    000075 AMOFAC     0004    000077 AMOMAX     0004    000076 AMOMIN     0004    000100 AMORAT
    C004    000054 AMQ        0004    000107 BETTER     0004    000102 BIAS       0003 R  000027 CIN        0005    000001 CONLV
    C004    000111 CORLEN     0003 R  000030 CTOI       0003 R  000041 DCON       0004    000053 DELT       0003 R  000043 DISS
    C004    000073 DWFAC      0004    000061 ELIMTH     0004    000052 EPS        0003 R  000050 GEN        0004    000074 GRACTM
    C003 P  000047 GREF       0000 I  000001 I          0003 I  000023 IDADJ      0003 I  000024 INDEX      0004    000066 INDXVL
    C000    000007 INJPS      0000 I  000003 J          0000 I  000004 JS         0003 I  000000 JUNK       0004    000051 KROOT
    C003 I  000020 LINK       0004    000003 LKURT      0000 I  000000 LOCA       0000 I  000005 LOCB       0004    000007 LOSUM
    C004    000004 LOVAR      0004 I  000002 LR         0004    000006 LSKEW      0003 I  000021 LSUBS      0004    000005 LSUM
    C003 I  000022 LSUPER     0004 I  000005 LV         0004    000002 LVRIN      0005 R  000012 MACCEL     0004    000001 MM
    C004    000110 MODE       0004 I  000000 MQ         0004 I  000011 MXAR       0003    000014 NARL       0004    000010 NINCLS
    C004    000070 NPTSO      0003 I  000043 NSYMB      0003 I  000020 NTB        0003    000016 NTBSZM     0003    000015 NTOP
    C003    000017 NWANT      0003 P  000045 OCIN       0004    000055 ODCON      0003 R  000047 ODEN       0003 R  000046 OPRIOR
    C003 R  000035 OPROP      0003 R  000036 OW         0005 R  000010 PACCEL     0003 R  000050 PCOND      0003 R  000025 PCUM
    C004    000103 PJOIN      0003 R  000043 PPASS      0003 R  000042 PORAT      0004    000071 PORATH     0003 R  000026 PRIRCM
    C003 P  000031 PROP       0003 R  000044 PST        0004    000065 SBLTH      0004    000062 SEPTH      0005    000002 SKBND
    C005    000003 SMCHI      0004    000112 SPCOR      0003 R  000032 SPFAC      0004    000072 SPMVTH     0000 R  000002 SUM
    C005    000004 TRBND      0005    000005 TRCHI      0005    000006 URKBND     0005    000007 URKCHI     0005 R  000014 VACCEL
    C004    000063 VFAC       0003 R  000037 VOLIN      0004    000101 VOLLIM     0003 R  000040 VOLRT      0003 R  000050 VRIN
    C004    000104 VRJOIN     0003 R  000034 W          0003 R  000033 WADJ       0004    000060 WADJIN     0005    000000 WAIT
    C004    000106 WDELSM     0004    000067 WFAC       0004    000105 WSIM       0004    000050 WTINIT     0004    000056 XOVFLO
    C004    000057 XUNFLO


00001  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*          SUBROUTINE VMTV(VA,AMET,VB)                                                  000010
00101      2*     C      SETS VA=AMET*VB                                                            000010
00102      3*          INCLUDE MISH                                                                 000010
00131      4*          REAL VA(30),VB(30),AMET(475)                                                 000010
00132      5*          LOCA=0                                                                       000010
00133      6*          DO 20 I=1,MQ                                                                 000016
00136      7*          SUM=0.                                                                       000021
```

```
00137      8*            DO 10 J=1,I                                          000026
00142      9*            LOCA=LOCA+I                                          000026
00143     10*         10 SUM=SUM+ AMET(LOCA)*VB(J)                            000030
00148     11*            IF(I.EQ.MO) GO TO 20                                 000037
00147     12*            JS=I+1                                               000042
00150     13*            LOCB=LOCA+I                                          000050
00151     14*            DO 11 J=JS,MO                                        000057
00154     15*            SUM=SUM+AMET(LOCB)*VB(J)                             000061
00158     16*         11 LOCB=LOCB+J                                          000065
00157     17*         20 VA(I)=SUM                                            000073
00161     18*            RETURN                                               000076
00162     19*            END                                                  000131

      END OF COMPILATION:       1 DIAGNOSTICS.
```

```
@FOR,S VPV,VPV,VPV
FOR SOE3-04/18/78-01:36:29 (0,0)

   SUBROUTINE VPV        ENTRY POINT 000027

   STORAGE USED: CODE(1) 000042; DATA(0) 000015; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    C003    CLUS    002017
    C004    MISC    000113
    C005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    C006    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    C001     000010 1336      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX
    C004     000076 AMOMIN    0004    000100 AMORAT   0004    000054 AMQ       0004    000107 BETTER    0004    000102 BIAS
    C003 R 000027 CIN         0005    000001 CONLV    0004    000111 CORLEN    0003 R 000030 CTOT      0003 R 000041 DCON
    C004     000053 DELT      0003 R 000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS
    C003 R 000050 GEN         0004    000074 GRACTM   0003 R 000047 GREF      0000 I 000000 I         0003 I 000023 IDADJ
    C003 I 000024 INDEX       0004    000066 INDXVL   0000    000002 INJP$     0003 I 000000 JUNK      0004    000051 KROOT
    C003 I 000020 LINK        0004    000003 LKURT    0004    000007 LOSUM     0004    000004 LOVAR     0004 I 000002 LR
    C004     000006 LSKEW     0003 I 000021 LSUB$     0004    000005 LSUM      0003 I 000022 LSUPER    0004 I 000005 LV
    C004     000002 LVRIN     0005 R 000012 MACCEL    0004    000001 MM        0004    000110 MODE      0004 I 000000 MQ
    C004 I 000011 MXAR        0003    000014 NARL     0004    000010 NINCLS    0004    000070 NPTS0     0003 I 000043 NSYMB
    C003 I 000020 NTB         0003    000016 NTBSZM   0003    000015 NTOP      0003    000017 NWANT     0003 R 000045 OCIN
    C004     000055 ODCON     0003 R 000047 ODEN      0003 R 000046 OPRIOR     0003 R 000035 OPROP     0003 R 000036 OW
    C005 R 000010 PACCEL      0003 R 000050 PCOND     0003 R 000025 PCUM       0004    000103 PJOIN     0003 R 000043 PPASS
    C003 R 000042 PQRAT       0004    000071 PQRATH   0003 R 000026 PRIRCM     0003 R 000031 PROP      0003 R 000044 PST
    C004     000065 SBLTH     0004    000062 SEPTH    0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR
    C003 R 000032 SPFAC       0004    000072 SPMVTH   0005    000004 TRBND     0005    000005 TRCHI     0005    000006 URKBND
    C005     000007 URKCHI    0005 R 000014 VACCEL    0004    000063 VFAC      0003 R 000037 VOLIN     0004    000101 VOLLIM
    C003 R 000040 VOLRT       0003 R 000050 VRIN      0004    000104 VRJOIN    0003 R 000034 W         0003 R 000033 WADJ
    C004     000060 WADJIN    0005    000000 WAIT     0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIM
    C004     000050 WTINIT    0004    000056 XOVFLO   0004    000057 XUNFLO

00000(  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*            SUBROUTINE VPV(VA,FAC,VB)                                          000010
00101      2*      C       SETS VA=VA*FAC*VB                                                000010
00103      3*            INCLUDE MISH                                                       000010
00131      4*            REAL VA(30),VB(30)                                                 000010
00132      5*            DO 10 I=1,MQ                                                       000010
00135      6*         10 VA(I)=VA(I)+VB(I)*FAC                                              000010
00137      7*            RETURN                                                             000014
00140(     8*            END                                                                000041
```